

Veritas High Availability Agent 5.0 for SAP NetWeaver Installation and Configuration Guide

AIX, HP-UX, Solaris

Veritas High Availability Agent 5.0 for SAP NetWeaver Installation and Configuration Guide

Copyright © 2006 Symantec Corporation. All rights reserved.

Veritas High Availability Agent 5.0 for SAP NetWeaver

Symantec and the Symantec logo are trademarks or registered trademarks of Symantec Corporation or its affiliates in the U.S. and other countries. Other names may be trademarks of their respective owners.

The product described in this document is distributed under licenses restricting its use, copying, distribution, and decompilation/reverse engineering. No part of this document may be reproduced in any form by any means without prior written authorization of Symantec Corporation and its licensors, if any.

THIS 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, SYMANTEC CORPORATION 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.

The Licensed Software and Documentation are deemed to be “commercial computer software” and “commercial computer software documentation” as defined in FAR Sections 12.212 and DFARS Section 227.7202.

Symantec Corporation
20330 Stevens Creek Blvd.
Cupertino, CA 95014
www.symantec.com

Third-party legal notices

Third-party software may be recommended, distributed, embedded, or bundled with this Symantec product. Such third-party software is licensed separately by its copyright holder. All third-party copyrights associated with this product are listed in the accompanying release notes.

AIX is a registered trademark of IBM Corporation.

HP-UX is a registered trademark of Hewlett-Packard Development Company, L.P.

Solaris is a trademark of Sun Microsystems, Inc.

Technical support

For technical assistance, visit <http://support.veritas.com> and select phone or email support. Use the Knowledge Base search feature to access resources such as TechNotes, product alerts, software downloads, hardware compatibility lists, and our customer email notification service.

Contents

Chapter 1	Introducing the Veritas Agent for SAP NetWeaver	
	What's new in this agent	7
	Supported software	8
	About the agent	8
	Agent operations	9
	Online operation	9
	Offline operation	9
	Monitor operation	10
	Clean operation	10
Chapter 2	Installing the Veritas Agent for SAP NetWeaver	
	Prerequisites for installing the agent for SAP NetWeaver	13
	Prerequisites for installing the agent in a VCS environment	13
	About the ACC library	14
	Upgrading the Veritas agent for SAP NetWeaver	14
	Installing the agent in a VCS environment	15
Chapter 3	Configuring the Veritas Agent for SAP NetWeaver	
	Importing the agent types files	17
	Agent attributes	18
	Required attributes	19
	Optional attributes	21
	Uniquely identifying SAP server instances	22
	Monitoring a SAP instance	23
	Setting the SAPMonHome attribute	24
	Executing a customized monitoring utility	24
	Preventing early faulting of Java and Add-In instances	25
Chapter 4	Clustering SAP Servers	
	Overview of the clustering process	27
	Scenario 1: Active-Active failover	29
	Scenario 2: Active-Passive failover	30
	Scenario 3: Enqueue Server failover	30
	Configuring SAP server instances for cluster support	32

Synchronizing accounts and services	32
Removing physical host dependencies from environment setup scripts	32
Removing physical host dependencies from profile file names	34
Removing physical host dependencies from profiles	34
Configuring SAPNW04 preonline script	35
Clustering shared file systems	37
For Oracle databases only	37
For non-Oracle databases	38
For other application servers	38

Chapter 5 Uninstalling the Veritas Agent for SAP NetWeaver

Chapter 6 Troubleshooting the Veritas Agent for SAP NetWeaver

Starting the SAP server outside a cluster server	41
Using correct software and operating system versions	44
Meeting prerequisites	44
Configuring SAP server resources	44
Checks for a SAP Java Add-In instance	44
Reviewing log files	45
Using agent log files	45
Using VCS log files	46
Using SAP server log files	46
Using trace level logging	47
Using trace level logging for preonline trigger	47

Appendix A Sample Configurations

Sample agent type definition	49
Sample SAP resource configuration	50
Sample SAP Central instance	50
Sample SAP Dialog instance	50
Sample SAP ABAP Enqueue Server instance	50
Sample SAP Java Enqueue Server instance	51
Sample SAP ABAP Enqueue Replication Server instance	51
Sample SAP Java Enqueue Replication Server instance	52
Sample service group configuration for ABAP and Java architectures	52
Sample service group configuration for Java Add-In architecture	57

Introducing the Veritas Agent for SAP NetWeaver

Veritas High Availability Agents monitor specific resources within an enterprise application, determine the status of these resources, and start or stop them according to external events. The Veritas High Availability Agent for SAP NetWeaver provides high availability for SAP R/3 and SAP NetWeaver in a cluster.

This guide assumes the user understands the primary components and basic functionality of a cluster. The guide also assumes a basic understanding of SAP R/3 and SAP R/3-based application architecture and configuration.

The agent for SAP NetWeaver is designed to support a wide range of SAP environments, including the traditional Basis architecture and the SAP J2EE Web Application Server architecture (NetWeaver). The agent also supports standalone Enqueue servers in a distributed SAP installation.

What's new in this agent

- Added support for all SAP NetWeaver architectures—ABAP, Java, and Java Add-In.
- Added support for Enqueue Replication server for all SAP NetWeaver architectures—ABAP, Java, and Java Add-In.

Supported software

The Veritas High Availability Agent for SAP NetWeaver support the following environments:

Environment	Supported Versions
Veritas Cluster Server	4.0, 4.1, 5.0
ACC Library	5.0
Operating Systems	Solaris—8, 9 HP-UX—11i v2 AIX—5L on pSeries
SAP R/3	4.6C with a 4.6D Kernel 4.6D 4.7 Enterprise Version
SAP Web AS	6.20
SAP NetWeaver	6.40

About the agent

The Veritas High Availability Agent for SAP NetWeaver brings SAP instances online, monitor 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 SAP instances are:

- Central instance
- Dialog instance
- Standalone Enqueue Servers
- Replication Enqueue Servers

The agent supports the following SAP Web Application Server architectures:

- ABAP
- Java
- Java Add-In

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

Agent operations

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

Online operation

The online operation performs the following tasks:

- Performs a preliminary check to ensure that the SAP instance is not `ONLINE` on the specified node in the cluster.
- Forcibly removes any SAP processes that remain because of an unclean shutdown.
 - If the SAP instance is a Central or Dialog instance, the `cleanipc` utility is executed. Otherwise, the agent simply kills all relevant SAP processes.
 - If the `kill.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.
- Initiates the standard SAP error log process.
- Starts the SAP O/S Collector that pipes the standard output and standard error messages to the SAP error log file.
- Starts the SAP instance using the `sapstart` command.
- Ensures that the instance is initialized successfully.

Offline operation

The offline operation performs the following tasks:

- Sends a `SIGINT` signal to the `sapstart` process, if the process exists. Otherwise, the operation 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` 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 a Central or Dialog instance, the operation executes the `cleanipc` utility.
- The operation also augments the SAP log.

Monitor operation

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

- Depending upon the search criteria that the [ProcMon](#) attribute specifies, the monitor operation scans the process table to verify the SAP instance processes are running.
 For more information about setting the ProcMon attribute, refer to [“Monitoring a SAP instance”](#) on page 23.
- If the [SecondLevelMonitor](#) attribute is greater than 0, the monitor operation performs a more thorough check of the SAP instance. The utility used to perform this second level check are:

Server architecture	Instance type	SAP utility used
SAP Web Application Server as ABAP	Central or Dialog	<code>sapinfo</code>
SAP Web Application Server as Java	Central or Dialog	<code>jcmmon</code>
SAP Web Application Server as Java Add-In	Central or Dialog	<code>sapinfo</code> and <code>jcmmon</code>

In case of an Enqueue Server or Enqueue Replication Server instances, the operation uses the `ensmon` utility.

- The monitor operation can also execute a custom monitor utility that the `MonitorProgram` attribute specifies. Refer to [“Executing a customized monitoring utility”](#) on page 24 for more information.

Clean operation

The clean operation performs the following tasks:

- Sends a `SIGINT` signal to the `sapstart` process, if the process exists. Otherwise, the operation sends a `SIGINT` signal to all running processes that are relevant to the specified SAP instance.
- Waits for 10 seconds for the SAP instance to shut down 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` 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 a Central or Dialog instance, the operation executes the `cleanipc` utility.
- The operation also augments the SAP log.

Installing the Veritas Agent for SAP NetWeaver

This chapter describes how to install the Veritas High Availability Agent for SAP NetWeaver in a cluster. You must install the agent on all systems that host a SAP service group.

Prerequisites for installing the agent for SAP NetWeaver

Ensure that you meet the prerequisites before installing the Veritas High Availability Agent for SAP NetWeaver.

Prerequisites for installing the agent in a VCS environment

- Install and configure Veritas Cluster Server.
- If you are working in a VCS 4.x environment, install the ACC library 5.0 (VRTSacc1ib) if it is not already installed. For more information, refer to [“About the ACC library”](#) on page 14.
- Remove any prior version of this agent.

About the ACC library

The operations for Veritas High Availability Agent for SAP NetWeaver depend on a set of PERL modules known as the ACC library. The library must be installed on each system in the cluster that will run the agent for SAP NetWeaver. The ACC library contains common, reusable functions that perform tasks such as process identification, logging, and system calls.

Note: If you are installing the agent for SAP NetWeaver in a VCS 5.0 environment, do not install the ACC library package separately. If you are installing the agent in a VCS 4.x environment, you must install the ACC library package before installing the agent.

To install or update the ACC library package, locate the library and related documentation either on the agent CD or in the compressed agent tar file.

Upgrading the Veritas agent for SAP NetWeaver

To upgrade the agent, first remove the older version of the agent. Refer to [“Uninstalling the Veritas Agent for SAP NetWeaver”](#) on page 39 for the uninstallation procedure. Then follow the instructions below to install the new agent software.

Installing the agent in a VCS environment

Perform the following steps on each system in the cluster.

To install the agent on AIX systems

- 1 Log in as root.
- 2 Go to the
`<cd_mount>/aix/application/sapnw_agent/<vcs_version>/pkgs` directory.
- 3 Install the package:

```
# installp -ac -d VRTSsapnw04.rte.bff VRTSsapnw04.rte
```

To install the agent on HP-UX systems

- 1 Log in as root.
- 2 Go to the
`<cd_mount>/hpux/application/sapnw_agent/<vcs_version>/pkgs` directory.
- 3 Install the package:

```
# swinstall -s `pwd` VRTSsapnw04
```

To install the agent on Solaris systems

- 1 Log in as root.
- 2 Go to the
`<cd_mount>/solaris/sparc/application/sapnw_agent/<vcs_version>/pkgs` directory.
- 3 Install the package:

```
# pkgadd -d . VRTSsapnw04
```


Configuring the Veritas Agent for SAP NetWeaver

After installing the agent for SAP NetWeaver, you must import the configuration file. After importing the configuration file, you can create and configure a SAP Server resource. Before you configure a resource, review the attributes table that describes the SAP resource type and its attributes.

To view sample agent type definition files and configuration service groups, refer to “[Sample Configurations](#)” on page 49.

Importing the agent types files

To use the agent for SAP NetWeaver, you must import the `SAPNW04Types.cf` file into the cluster.

To import the agent types files to work with VCS

Perform the following steps using the VCS graphical user interface.

- 1 Start the Veritas Cluster Manager GUI 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:

Version Directory structure

VCS 4.x `/etc/VRTSvcs/conf/sample_SAPNW04/SAPNW04Types.cf`

VCS 5.0 `/etc/VRTSagents/ha/conf/SAPNW04/SAPNW04Types.cf`

- 4 Click **Import**.

5 Save the VCS configuration.

The SAP configuration is now imported to the VCS engine. You can now create SAP Server resources. For additional information about using the VCS GUI, refer to the *Veritas Cluster Server User's Guide*.

Refer to “[Sample agent type definition](#)” on page 49 for sample agent definition.

Agent attributes

The required and optional attributes required for configuring a SAP instance are described below.

Required attributes

Required attributes	Description
EnqSrvResName <i>String</i>	The name of the standalone Enqueue server cluster resource. An Enqueue Replication server uses this attribute to allow the Replication server to query the Enqueue server resource state. The Enqueue Replication server monitors the Enqueue server state to determine if a Replication server fail over is necessary. Example: ASCS02_srv Default: ""
EnvFile <i>String</i>	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 <code>ksh</code> , <code>sh</code> , and <code>csh</code> . Note: Ensure that the syntax of this file is in accordance with the user shell that the SAPAdmin attribute specifies. Symantec recommends that you store this file on shared disk so that the file is always available to an online system. Example: /usr/sap/EA1/DVEBMGS00/envfile Default: ""
InstName <i>String</i>	Identifies a SAP server instance. For more information about setting this attribute, refer to " Uniquely identifying SAP server instances " on page 22. Example: DVBGS01 Default: DVEBMGS00
InstType <i>String</i>	An identifier that classifies and describes the SAP server instance type. Valid values are: <ul style="list-style-type: none"> ■ CENTRAL: SAP Central Services instance ■ DIALOG: SAP Dialog instance ■ ENQUEUE: SAP Standalone Enqueue instance ■ AENQUEUE: ABAP SAP Standalone Enqueue instance ■ JENQUEUE: Java SAP Standalone Enqueue instance ■ ENQREP: SAP Enqueue Replication Server ■ AENQREP: ABAP SAP Enqueue Replication Server ■ JENQREP: Java SAP Enqueue Replication Server Note: The value of this attribute is not case sensitive. Example: DIALOG Default: CENTRAL

Required attributes	Description
ProcMon <i>String</i>	<p>The list of SAP processes to monitor. The entries in this list are separated using space and can be specified in any order.</p> <p>For more information about how the monitor operation uses this attribute, refer to “Monitoring a SAP instance” on page 23.</p> <p>Example: dw se jc</p> <p>Default: ““</p>
ResLogLevel <i>String</i>	<p>The logging detail performed by the agent for the resource. Valid values are:</p> <p>ERROR: Only logs error messages.</p> <p>WARN: Logs above plus warning messages.</p> <p>INFO: Logs above plus informational messages.</p> <p>TRACE: Logs above plus trace messages. TRACE is very verbose and should only be used during initial configuration or for troubleshooting and diagnostic operations.</p> <p>Example: TRACE</p> <p>Default: INFO</p>
SAPAdmin <i>String</i>	<p>UNIX user that is defined as the administrator of application directories and executables. This user name is usually a concatenation of the SAPSID attribute and the <code>adm</code> string.</p> <p>This user name is stored in one or more system naming services, for example, NIS, NIS+, and LDAP servers. The agent operations use this user name to execute their respective core subroutines.</p> <p>Example: ea1adm</p> <p>Default: ““</p>
SAPMonHome <i>String</i>	<p>The location of the directory that contains the binary used for second level monitoring process. For more information about setting the SAPMonHome attribute, refer to “Setting the SAPMonHome attribute” on page 24.</p> <p>Example: /usr/sap/\${SAPSID}/SYS/exe/runU</p> <p>Default: ““</p>
SAPSID <i>String</i>	<p>SAP system name.</p> <p>The value of this attribute is exactly 3 characters in length, and must begin with an alphabet. The value of this attribute is defined during the SAP installation.</p> <p>For more information about setting this attribute, refer to “Uniquely identifying SAP server instances” on page 22.</p> <p>Example: EA1</p> <p>Default: ““</p>

Required attributes	Description
StartProfile <i>String</i>	The full path and file name of the StartProfile instance. The StartProfile instance is found in <code>/usr/sap/<SAPSID>/SYS/profile</code> directory. The value of the instance is <code>START_<InstName>_<hostname></code> . The <code><hostname></code> must resolve into a valid IP address that is used to cluster the SAP instance. Example: <code>/usr/sap/EA1/SYS/profile/START_DVEBMGS01_sunabap</code> Default: ""

Optional attributes

Optional attribute	Description
MonitorProgram <i>String</i>	Absolute path name of an external, user-supplied monitor executable. For information about setting this attribute, refer to: <ul style="list-style-type: none"> ■ “Executing a customized monitoring utility” on page 24 ■ “Preventing early faulting of Java and Add-In instances” on page 25 Example 1: <code>/usr/sap/EA1/DVEBMGS00/work/myMonitor.sh</code> Example 2: <code>/usr/sap/EA1/DVEBMGS00/work/myMonitor.sh arg1 arg2</code> Default: ""
SecondLevelMonitor <i>Integer</i>	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 <code>sapinfo.exe</code> is executed every 100 minutes, which may not be as often as intended. For maximum flexibility, no upper limit is defined for SecondLevelMonitor. Example: 1 Default: 0

Uniquely identifying SAP server instances

You can *virtualize* a SAP instance using a cluster. Using shared disk and virtual IP addresses, you can manage a large set of SAP instances in a single cluster. In cases such as multiple instances running concurrently on a single node, the agent must be able to uniquely identify each SAP instance on a single host system.

Each instance have a unique instance name. The instance names may follow the conventional form. For example, Dialog instances begin with 'D', and Central instances are typically named DVEBMGS.

Instance names often include an instance ID suffix which is an integer between 00-96. For example, a Central instance with an instance ID = 01 may have an instance name of DVEBMGS01.

Note that the [SAPSID](#) and [InstName](#) attributes form a unique identifier that can identify the processes running for a particular instance.

Some examples are:

SAP instance name	SAP instance type
DVEBMGS01	SAP BASIS Central instance - ABAP
DVBGS01	SAP BASIS Central instance without Enqueue and Msg Servers
JC00	SAP Central instance - Java
ASCS00	SAP Standalone Enqueue Server
SCS00	SAP Central Services
D01	SAP Dialog instance- ABAP
J02	SAP Dialog instance- Java
REP03	SAP Enqueue Replication Server

Differentiating SAP instances is important when the agent must kill 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 a SAP instance

The monitor operation performs process level check to ensure the proper functioning of a 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.

Valid values of the ProcMon attribute follow.

SAP architecture	SAP instance type	Value of Procmon attribute
ABAP	Central	dw se co
	Dialog	dw se co
	Enqueue Server	en ms se co
	Enqueue Replication Server	er enr
Java	Central	jc
	Dialog	jc
	Enqueue Server	en ms
	Enqueue Replication Server	er enr
Java Add-In	Central	dw jc co and se are optional.
	Dialog	For ABAP: dw For Java: jc
	Enqueue Server	en ms
	Enqueue Replication Server	er enr

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.

Setting the SAPMonHome attribute

The [SAPMonHome](#) attribute contains the absolute path to the directory that contains the binary used for second level monitoring process. The binaries used are:

- For ABAP: `sapinfo`
- For Java: `jcmon`
- For Java Add-In: `sapinfo, jcmon`

Recommended values for the SAPMonHome attribute are:

SAP architecture and instance	Format	Value of the SAPMonHome attribute
SAP ABAP For all instances	Unicode and Non-Unicode	<code>/usr/sap/<SAPSID>/SYS/exe/run</code>
SAP Java For all instances	Unicode	<code>/usr/sap/<SAPSID>/SYS/exe/run</code>
SAP Java Add-In When <code>InstType</code> is equal to CENTRAL, DIALOG, AENQUEUE, or AENQREP.	Unicode and Non-Unicode	<code>/usr/sap/<SAPSID>/SYS/exe/run</code>
SAP Java Add-In When <code>InstType</code> is equal to JENQUEUE or JENQREP.	Unicode and Non-Unicode	<code>/usr/sap/<SAPSID>/SYS/exe/runU</code>

Executing a customized monitoring utility

The monitor operation can execute a customized monitoring utility to perform an additional SAP server state check. The monitor operation executes the utility specified in the [MonitorProgram](#) attribute, if:

- The specified utility is a valid executable file.
- The first level process check indicates that the SAP Server instance is online.
- The `SecondLevelMonitor` attribute is either set to 0 or 1, and the second level check indicates that the SAP 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 operation interprets the utility exit code as follows:

Exit code	Interpretation
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, Symantec recommends storing the file in a shared directory that is available on the online node.

Preventing early faulting of Java and Add-In instances

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

This agent for SAP NetWeaver 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.

Symantec 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 `jc` process, you must set the value of the `MonitorTimeout` attribute to at least 90 seconds.

Clustering SAP Servers

The SAP Central instances and databases are single points of failure, and you need to protect these instances from hardware and software failures.

This chapter describes configurations that protect the SAP instances from hardware and software failures. This chapter also covers steps to configure SAP instances for successful cluster support.

Overview of the clustering process

A cluster must have spare capacity to handle SAP instance failover scenarios. For example, in case of a backend database failure, the cluster must be able to run another database instance, in conjunction with another running applications. The *Veritas Cluster Server Agent Installation and Configuration Guide* covers details about database protection.

A cluster can provide application failover by encapsulating the resources required for an application into a *service group*. A service group is a *virtualized* application that can move among cluster nodes. A service group can contain a set of dependent resources, such as disk groups, disk volumes, file systems, IP addresses, NIC cards, and dependent application processes.

A cluster can start, stop, monitor, and switch service groups within the cluster, depending upon server or resource faults. An administrator can proactively move a service group between cluster nodes to perform preventative maintenance or apply patches. The service group includes logic about the dependencies between application components.

The configurations described here are based on the SAP-recommended switchover scenarios described in the SAP white paper, *SAP Web Application Server in Switchover Environments (UNIX Platforms, Release 6.40, February 2006)*. This paper describes six SAP failover scenarios. You can configure the

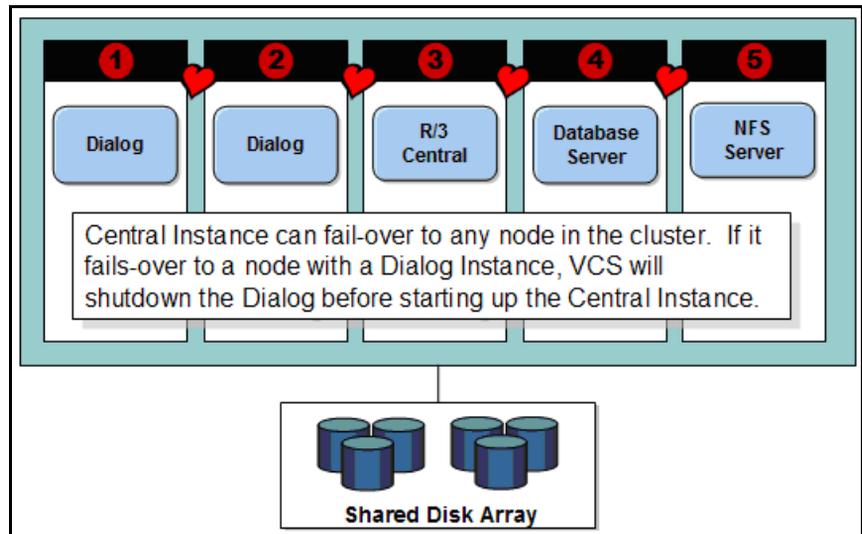
cluster to address all six failover scenarios. A summary of these scenarios follows.

Scenario	Normal Operation				After Switchover			
	Switchover Cluster		External Nodes		Switchover Cluster		External Nodes	
	Node C1	Node C2	Node E1	Node E2	Node C1	Node C2	Node E1	Node E2
SO-1	CI	DB	—	AS	<i>failed</i>	DB+CI	—	AS
SO-2	DB	CI	—	AS	<i>failed</i>	CI+DB	—	AS
SO-3	CI/DB	AS	—	AS	<i>failed</i>	CI/DB	—	AS
SO-4	CI	AS	DB	AS	<i>failed</i>	CI	DB	AS
SO-5	DB	AS	CI	AS	<i>failed</i>	DB	CI	AS
SO-6	DB	<i>Idle</i>	CI	AS	<i>failed</i>	DB	CI	AS

The Veritas High Availability Agent for SAP NetWeaver supports all these SAP scenarios. Samples of commonly implemented switchover scenarios, that are also SAP recommended best practices are described below.

Scenario 1: Active-Active failover

Consider a database and a SAP Central instance placed on different cluster nodes. If the host running the database fails, the database automatically switches to any other node in the cluster. Similarly, if the host running the SAP Central instance fails, the instance can switch to any other node in the cluster, and resume operability.



Caution: If you are using a two-node cluster to host a SAP Central instance, ensure that you properly provision the machines the machine resources such as CPU, memory, and so on. In case of a failover, the machines must be able to run the database and SAP Central instance together.

Typically, the database and the Central instance have their own service groups. In this example, the instance is on system 3 and the database is on system 4. Because the database must be running before the Central instance can be started, the cluster brings the database service group online before the Central instance is started. If DRGRP is the database service group and CIGRP is the instance service group, run this command to establish the appropriate group dependency:

```
# hagr -link DBGRP CIGRP online global
```

Note: DBGRP must be online in a cluster node, before CIGRP comes online.

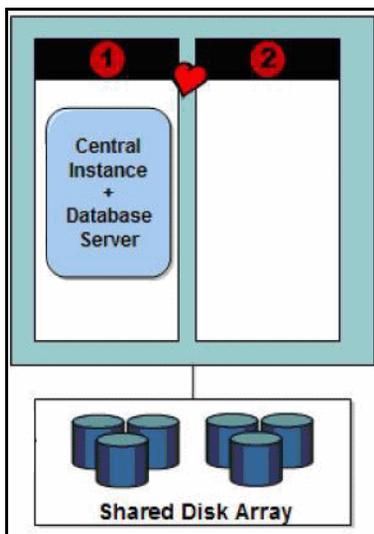
Refer to “[Sample service group configuration for ABAP and Java architectures](#)” on page 52 for a sample service group configuration that shows an active-active failover scenario.

Scenario 2: Active-Passive failover

In this example, a two-node cluster is setup using an active-passive design. The active node is configured to run both the database and SAP Central instance. In case of an active node failure, the cluster performs an orderly shutdown of the active node, followed by an orderly start of the passive node.

Caution: Dialog servers may see a temporary delay in the Central instance availability, but can soon reconnect and re-establish locks to continue transaction processing.

A diagram to illustrate a two-node cluster configuration follows.



Refer to “[Sample service group configuration for ABAP and Java architectures](#)” on page 52 for a sample service group configuration that shows an active-passive failover scenario.

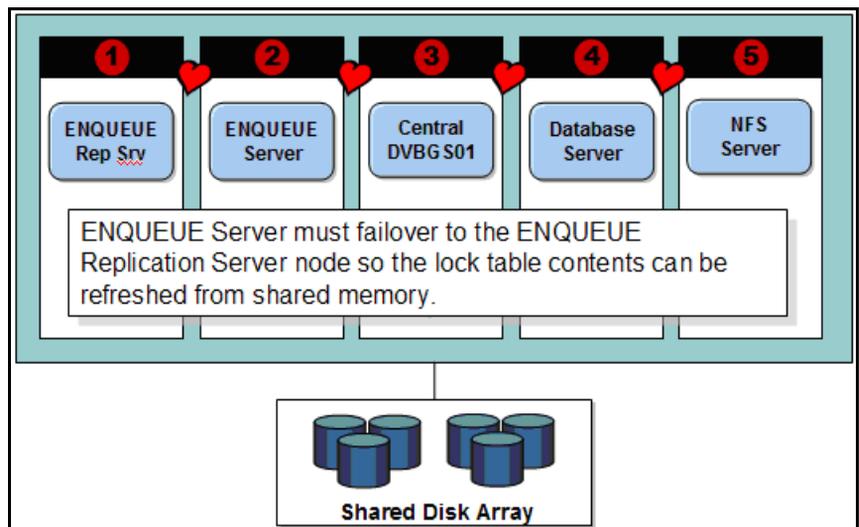
Scenario 3: Enqueue Server failover

In large and mission critical SAP installations, you can choose to split the enqueue lock service into a separate service group, and can configure a

replication engine to maintain lock table state across cluster boundaries. Therefore, you can provide a standalone Enqueue server and associated Replication servers to ensure high availability in mission critical environment. For more information about this SAP high availability feature, refer to the SAP white paper, *The SAP Lock Concept (BC-CST-EQ - Release 640)*.

In this scenario, SAP Central instance, Enqueue server, Enqueue Replication server, and the database are configured in separate service groups. These resources may run on the same physical node or on separate nodes. A diagram to illustrate this configuration follows.

Caution: The Enqueue server and the Replication server must run on separate physical nodes, to ensure lock state preservation in case of an Enqueue server failover.



This configuration is to provide fault tolerance for the Enqueue lock table. For example, if node 2 experiences a failure, the Enqueue server running on node 2 must failover to cluster node 1. Upon restart, the Enqueue server can read the node 1 shared memory contents to refresh the lock table.

Once the lock transfer state returns to '0' (return code from the `ensmon` program), the Enqueue Replication server faults and migrates to another node in the cluster. The server can once again begin buffering the primary lock state information that the Enqueue server generates on cluster node 1.

This failover behavior is a core feature of the agent for SAP NetWeaver. This process automatically occurs in case a SAP site experiences an Enqueue server cluster node failure.

Refer to “[Sample service group configuration for ABAP and Java architectures](#)” on page 52 for a sample service group configuration that shows an active-active with Enqueue server scenario.

Configuring SAP server instances for cluster support

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

Synchronizing accounts and services

Synchronize user and group accounts:

- 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:

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

Removing physical host dependencies from environment setup scripts

You can create a node-independent set of script files that can setup the SAP environment. You can run these files on each node in the cluster to setup identical SAP environment on all nodes.

Caution: Verify that the Bourne shell and C-shell versions on each node are identical.

An example of a Bourne shell script to create such script files for a SAP APO module follows.

```
#!/bin/sh
# -----
```

```

# Setup the environment file list and
# determine the name of the local system
# -----
ENVFILELIST=".sapenv .apoenv .j2eeenv .dbenv"
LOCALHOSTNAME=`uname -n`
# -----
#   Create the backup directory
# -----
if [ ! -d ./backup ]
then
    echo "Create backup Directory"
    mkdir ./backup
fi
# -----
# loop thru the env file list and
# copy the environment setup file making
# a backup of the original
# -----
for i in $ENVFILELIST
do
    FILENAME="./${i}_${LOCALHOSTNAME}.sh"
    if [ -f ${FILENAME} ]
    then
        echo "Making backup of $FILENAME"
        cp $FILENAME ./backup/$FILENAME
        echo "Changing file name to be host independent"
        mv $FILENAME ${i}_ALLHOSTS.sh
    fi
done

```

You can also create a master environment setup script file that sources all separate setup scripts, such as `.sapenv`, `.apoenv`, `.j2eeenv`, `.dbenv`, and so on. You can configure the `EnvFile` attribute so that each agent operation uses this master script to setup the SAP environment.

A script to configure SAPAdmin accounts to use the Bourne shell as default, is provided as an example below.

```

# -----
# Set the environment file list
# Determine the OS
# -----
ENVFILELIST=".sapenv .apoenv .j2eeenv .dbenv"
# -----
# loop thru the env file list and source
# the os dependent env file
# -----
for i in $ENVFILELIST
do
    FILENAME="${HOME}/${i}_ALLHOSTS.sh"
    if [ -f $FILENAME ]
    then
        . $FILENAME
    fi
done

```

```
fi  
done
```

Removing physical host dependencies from profile file names

Ensure that the start and instance profile names in `/usr/sap/<SAPSID>/SYS/profile` append the virtual host names, instead of machine dependent host name values.

An example of a profile directory populated with start and instance profiles, appended with virtual host names follows.

```
bash-2.05# ls -l  
total 48  
-rw-r--r--  1 ecladm  sapsys      611 Mar 23 17:26 DEFAULT.PFL  
-rw-r--r--  1 ecladm  sapsys      835 Mar 30 17:33  
JAI_ASCS02_sunascs  
-rw-r--r--  1 ecladm  sapsys      308 Feb 25 19:46  
JAI_DVBGS01_sunabap  
-rw-r--r--  1 ecladm  sapsys      310 Feb 23 20:16  
JAI_DVEBMGS01_sunabap  
-rw-r--r--  1 ecladm  sapsys      670 Feb 25 19:47  
JAI_REP02_sunreps  
-rw-r--r--  1 ecladm  sapsys     2571 Mar 24 15:03  
START_ASCS02_sunascs  
-rw-r--r--  1 ecladm  sapsys     1798 Mar 24 15:47  
START_DVBGS01_sunabap  
-rw-r--r--  1 ecladm  sapsys     2427 Feb 23 20:16  
START_DVEBMGS01_sunabap  
-rw-r--r--  1 ecladm  sapsys     1150 Mar 24 15:21  
START_REP02_sunreps
```

Removing physical host dependencies from profiles

- Edit the start profile
`/sapmnt/<SAPSID>/profile/START_<InstName>_<VirtualHostName>` to replace all physical hostnames with their equivalent virtual hostname.
- Edit the instance profile
`/sapmnt/<SAPSID>/profile/<SAPSID>_<InstName>_<VirtualHostName>`:
 - a Add these lines:
 - `SAPGLOBALHOST = virtualhostname`
 - `SAPLOCALHOST = virtualhostname`
 - `SAPLOCALHOSTFULL = virtualhostname`
 - b Replace all physical hostnames with the equivalent virtual host name.

- Edit the default profile
`/usr/sap/<SAPSID>/SYS/profile/DEFAULT.PFL:`
 - a Set SAPDBHOST equal to the virtual hostname of the database server.
 - b Replace all instances of the physical hostname of the SAP Central instance, with the equivalent virtual host name.
 - c Verify that all physical hostnames are replaced with equivalent virtual hostnames.

Configuring SAPNW04 preonline script

In a clustered environment, the SAP administrator installs and configures the SAP standalone Enqueue and SAP Enqueue Replication server. The behavior of these Enqueue servers has two requisites:

- If a standalone Enqueue server instance fails, the server must fail over to the node where the Enqueue Replication server instance is online. In the Enqueue Replication server instance, once the lock table state is replicated, the Replication server instance must switch to another node in the cluster.
- If the Enqueue Replication server instance fails, the instance must fail over to such a node in the cluster that does *not* have an Enqueue server instance in ONLINE state.

The SAPNW04 preonline script facilitates proper Enqueue server failover behavior. The existing VCS preonline script calls the SAPNW04 preonline script.

The SAPNW04 preonline script performs these tasks:

- If the service group in 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

- 1 For the service group, set the preonline flag to true:

```
# hagr -modify service_group preonline 1
```
- 2 Go to the `$VCS_HOME/bin/triggers` directory.
- 3 In the preonline file, add these lines to integrate the call to the SAPNW04 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 SAPNW04 Trigger Call here...
#-----
# Define variables..
#-----
my $sCmd = '/opt/VRTSvcs/bin/SAPNW04/preonline';
```

Note: For VCS 5.0, the value of \$sCmd must be equal to
/opt/VRTSagents/ha/bin/SAPNW04/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:
- a Pick the sample preonline script present in this directory:

VCS Version	Directory
4.x	/etc/VRTSvcs/conf/sample_SAPNW04
5.0	/etc/VRTSagents/ha/conf/SAPNW04

- b Copy this file in the `$VCS_HOME/bin/triggers` directory.
- c Ensure that the file is an executable, and the root user can access it.
- d Create a symlink for the preonline script to the monitor script by running these commands:

VCS Version	Creating a symlink
4.x	# cd /opt/VRTSvcs/bin/SAPNW04 # ln -s monitor preonline
5.0	# cd /opt/VRTSagents/ha/bin/SAPNW04 # ln -s monitor preonline

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`.

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 instance shares with the database or with other application servers.

For Oracle databases only

The SAP Central instance runs on a separate machine, while the Oracle database is mounted on `/oracle/<SAPSID>` by default. The database has its own `/oracle/<SAPSID>` file system, but also requires SAP executables. These

executables are usually NFS-mounted from `/sapmnt/<SAPSID>/exe` on the Central instance.

Symantec does not recommend that the Central instance and the database run on the same node. Symantec recommends:

- maintaining local copies of `/oracle/<SAPSID>` and `/sapmnt/<SAPSID>/exe` on the Central instance and the database, instead to sharing them through NFS.
- keeping database files on shared disk.
- controlling the `oracle/<SAPSID>` and `/sapmnt/<SAPSID>/exe` file systems through the operating system, instead of the cluster.

For non-Oracle databases

The database requires SAP executables. These executables are usually NFS-mounted from `/sapmnt/<SAPSID>/exe` on the Central instance.

Symantec does not recommend that the Central instance and the database run on the same node. Symantec recommends:

- maintaining local copies of `/oracle/<SAPSID>` and `/sapmnt/<SAPSID>/exe` on the Central instance and the database, instead to sharing them through NFS.
- keeping database files on shared disk.
- controlling the `oracle/<SAPSID>` and `/sapmnt/<SAPSID>/exe` file systems through the operating system, instead of the cluster.

For other application servers

The application servers require `/usr/sap/trans`, `/sapmnt/<SAPSID>/global`, and `/sapmnt/<SAPSID>/profile` to be NFS-mounted from the Central instance.

Therefore you must share these resources using NFS. But Symantec recommends that you 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, Version 6.3, October 2003)*.

Uninstalling the Veritas Agent for SAP NetWeaver

Follow the steps below to remove the Veritas High Availability agent for SAP NetWeaver from a cluster. These steps must be performed while the cluster is active.

To uninstall the agent in a VCS environment

- 1 Log in as `root`.
- 2 Set the cluster configuration mode to read/write by typing the following command from any system in the cluster:

```
# haconf -makerw
```
- 3 Remove all SAP resources from the cluster. Use the following command to verify that all resources have been removed.

```
# hares -list Type=SAPNW04
```
- 4 Remove the agent from the cluster configuration by typing the following command from any system in the cluster.

```
# hatype -delete SAPNW04
```

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 system in the cluster:

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

- 6 Use the platform's native software management program to remove the agent for SAP NetWeaver from each node in the cluster:

Platform	Command
AIX	# installp -u VRTSsapnw04.rte
HP-UX	# swremove VRTSsapnw04
Solaris	# pkgrm VRTSsapnw04

Troubleshooting the Veritas Agent for SAP NetWeaver

This chapter covers tips and pointers on using the agent for SAP NetWeaver, with Veritas high availability products. To resolve issues effectively, follow the steps in the order presented below. You may come across unique issues, but make sure that you follow these steps in the presented order to avoid unnecessary issues.

These troubleshooting tips and pointers are applicable to the agent while working in a VCS environment.

Starting the SAP server outside a cluster server

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 server independent of the cluster framework. Refer to the cluster documentation for information about disabling a resource.

You can then restart the SAP server outside the cluster framework.

Note: Use the same parameters that the resource attributes define within the cluster framework while restarting the resource outside the framework.

A sample procedure to start a SAP instance outside the cluster framework, is illustrated below.

To restart the SAP instance outside the framework

- 1 Log in as root.
- 2 Ensure that the SAP database is up and running. Refer to the relevant database documentaion 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=/lhome/<SAPAdmin>
$ export USER LOGNAME HOME
$ . <EnvFile>
```

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

- 4 Start the SAP server to run the instance:

```
$ sapstart pf=<StartProfile>
```

You may see the following logs:

```
SAP-R/3-Startup Program Rel 640 V1.8 (2003/04/24)
-----
Starting at 2006/05/04 13:01:35
Startup Profile:
"/usr/sap/<SAPSID>/SYS/profile/START_<InstName>_sunabap"

Execute Pre-Startup Commands
-----
(18439) Local: /sapmnt/<SAPSID>/exe/sapmscsa -n
pf=/usr/sap/<SAPSID>/SYS/profile/<SAPSID>_<InstName>_sunabap
/sapmnt/<SAPSID>/exe/sapmscsa: make new mode. SCSA currently non
existent.
sapcsca: SCSA defined. sapcscaId == 7948 == 00001f0c
sapcsca: SCSA attached at address ffffffff7cb00000
sapcsca: SCSA initialized.
rslgwrl(21): Searching for overlap point in pre-existing SysLog
file...
/sapmnt/<SAPSID>/exe/sapmscsa: finished.
(18439) Local: rm -f se.sap<SAPSID>_<InstName>
(18439) Local: ln -s -f /sapmnt/<SAPSID>/exe/rslgsend
se.sap<SAPSID>_<InstName>
(18439) Local: rm -f dw.sap<SAPSID>_<InstName>
(18439) Local: ln -s -f /sapmnt/<SAPSID>/exe/disp+work
dw.sap<InstName>

Starting Programs
-----
(18450) Starting: local se.sap<SAPSID>_<InstName> -F
pf=/usr/sap/<SAPSID>/SYS/profile/<SAPSID>_<InstName>_sunabap
(18451) Starting: local dw.sap<SAPSID>_<InstName>
pf=/usr/sap/<SAPSID>/SYS/profile/<SAPSID>_<InstName>_sunabap
(18439) Waiting for Child Processes to terminate.
```

- 5 Ensure that the SAP instance is running successfully by running the `grep` command for `<InstName>`.

For example, for a SAP Central instance:

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

You must see the following processes running on the system:

```
dw.sap<SAPSID>_<InstName>
pf=/usr/sap/<SAPSID>/SYS/profile/<SAPSID>_<InstName>_sunabap
<SAPAdmin> 18552 18451 0 13:01:45 ? 0:00
dw.sap<SAPSID>_<InstName>
pf=/usr/sap/<SAPSID>/SYS/profile/<SAPSID>_<InstName>_sunabap
<SAPAdmin> 18549 18451 1 13:01:45 ? 0:08
dw.sap<SAPSID>_<InstName>
pf=/usr/sap/<SAPSID>/SYS/profile/<SAPSID>_<InstName>_sunabap
<SAPAdmin> 18555 18451 0 13:01:45 ? 0:00
dw.sap<SAPSID>_<InstName>
pf=/usr/sap/<SAPSID>/SYS/profile/<SAPSID>_<InstName>_sunabap
<SAPAdmin> 18451 18439 0 13:01:36 ? 0:02
dw.sap<SAPSID>_<InstName>
pf=/usr/sap/<SAPSID>/SYS/profile/<SAPSID>_<InstName>_sunabap
<SAPAdmin> 18554 18451 0 13:01:45 ? 0:00
dw.sap<SAPSID>_<InstName>
pf=/usr/sap/<SAPSID>/SYS/profile/<SAPSID>_<InstName>_sunabap
<SAPAdmin> 18450 18439 0 13:01:36 ? 0:00
se.sap<SAPSID>_<InstName> -F
pf=/usr/sap/<SAPSID>/SYS/profile/<SAPSID>_<InstName>_sunabap
<SAPAdmin> 18551 18451 0 13:01:45 ? 0:00
dw.sap<SAPSID>_<InstName>
pf=/usr/sap/<SAPSID>/SYS/profile/<SAPSID>_<InstName>_sunabap
<SAPAdmin> 18556 18451 0 13:01:45 ? 0:00
dw.sap<SAPSID>_<InstName>
pf=/usr/sap/<SAPSID>/SYS/profile/<SAPSID>_<InstName>_sunabap
<SAPAdmin> 18547 18451 0 13:01:45 ? 0:01 gwrld -dp
pf=/usr/sap/<SAPSID>/SYS/profile/<SAPSID>_<InstName>_sunabap
<SAPAdmin> 18550 18451 0 13:01:45 ? 0:05
dw.sap<SAPSID>_<InstName>
pf=/usr/sap/<SAPSID>/SYS/profile/<SAPSID>_<InstName>_sunabap
<SAPAdmin> 18548 18451 0 13:01:45 ? 0:01 icman -attach
pf=/usr/sap/<SAPSID>/SYS/profile/<SAPSID>_<InstName>_sunabap
<SAPAdmin> 18439 1 0 13:01:36 ? 0:00 sapstart
pf=/usr/sap/<SAPSID>/SYS/profile/START_<InstName>_sunabap
```

- For a SAP Java instance, you must look for `jc` processes. For example:

```
jc.sap<SAPSID>_<InstName>
pf=/usr/sap/<SAPSID>/SYS/profile/<SAPSID>_<InstName>_sunj2e
```

- For a SAP Java Add-In instance, you must specify `dw jc` in the value of the [ProcMon](#) attribute. Then, you can look for `jcontrol` processes to ensure that the instance is running. For example:

```
jcontrol
pf=/usr/sap/<SAPSID>/SYS/profile/<SAPSID>_<InstName>_sunabap
```

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.

Using correct software and operating system versions

To ensure that no issues arise due to incorrect software and operating system versions, refer to “[Supported software](#)” on page 8 for the correct versions of operating system and software to be installed on the resource systems.

Meeting prerequisites

Before installing the agent for SAP NetWeaver, double check that you meet the prerequisite requirements. For example, you must install the ACC library on VCS before installing the agent. Refer to “[Prerequisites for installing the agent for SAP NetWeaver](#)” on page 13 for the list of prerequisites.

Configuring SAP server resources

Before using a SAP server resource, ensure that you configure the resource properly. Refer to “[Agent attributes](#)” on page 18 for the list of resource types with which you can configure all SAP server resources.

Checks for a SAP Java Add-In instance

For a SAP Java Add-In instance, you must perform the following checks before further investigations:

- The SAP resources running the ABAP and Java Standalone Enqueue server instances, must be in the same Service Group as the SAP Central instance.
- The SAP resources running the ABAP and Java Enqueue Replication server instances, must be in the same Service Group.
- For the Standalone Enqueue server instances, the value of the [InstType](#) attribute must not be **ENQUEUE**. The values must be:
 - For ABAP: **AENQUEUE**
 - For Java: **JENQUEUE**
- For the Enqueue Replication server instances, the value of the InstType attribute must not be **ENQREP**. The values must be:

- For ABAP: **AENQREP**
- For Java: **JENQREP**
- Ensure that:
 - The **EnqSrvResName** attribute of the Java Enqueue Replication server instance is set to the VCS resource that is running the corresponding Java Standalone Enqueue server instance.
 - The **EnqSrvResName** attribute of the ABAP Enqueue Replication server instance is set to the VCS resource that is running the corresponding ABAP Standalone Enqueue server instance.

Reviewing log files

If you are facing problems while using the agent for SAP NetWeaver or a SAP server instance, refer to the following sections to access the relevant files for information about the issue.

Using agent log files

In case of problems while using the agent for SAP NetWeaver, you can access the agent log files 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`.

For example, if `<SAPSID>` is `EC1`, some valid log files can be:

SAP architecture	Instance type	Agent log file
ABAP	Central instance, without replication configuration	<code>/usr/sap/EC1/DVEBMGS01/log/EC1_DVEBMGS01.log</code>
	Central instance, with replication configuration	<code>/usr/sap/EC1/DVBGS01/log/EC1_DVBGS01.log</code>
	Dialog instance	<code>/usr/sap/EC1/D03/log/EC1_D01.log</code>
	Standalone Enqueue server instance	<code>/usr/sap/EC1/ASCS02/log/EC1_ASCS02.log</code>
	Enqueue Replication server instance	<code>/usr/sap/EC1/REP02/log/EC1_REP02.log</code>

SAP architecture	Instance type	Agent log file
Java	Central instance	/usr/sap/EC1/JC01/log/EC1_JC01.log
	Dialog instance	/usr/sap/EC1/J01/log/EC1_J01.log
	Standalone Enqueue server instance	/usr/sap/EC1/SCS04/log/EC1_SCS04.log
	Enqueue Replication server instance	/usr/sap/EC1/REP04/log/EC1_REP04.log
Java Add-In	Central instance	/usr/sap/EC1/DVEBMGS01/log/EC1_DVEBMGS01.log
	Dialog instance (ABAP)	/usr/sap/EC1/J01/log/EC1_J01.log
	Dialog instance (Java)	/usr/sap/EC1/D01/log/EC1_D01.log
	Standalone Enqueue server instance (ABAP)	/usr/sap/EC1/SCS05/log/EC1_SCS04.log
	Enqueue Replication server instance (ABAP)	/usr/sap/EC1/REP05/log/EC1_REP04.log
	Standalone Enqueue server instance (Java)	/usr/sap/EC1/ASCS06/log/EC1_SCS04.log
	Enqueue Replication server instance (Java)	/usr/sap/EC1/REP06/log/EC1_REP04.log

Using VCS log files

In case of problems while using the agent for SAP NetWeaver, you can also access the VCS engine log file for more information about a particular resource. The VCS engine log file is `/var/VRTSvcs/log/engine_A.log`.

Using SAP 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.

Using trace level logging

The [ResLogLevel](#) attribute controls the level of logging that is written in a VCS log file for each SAP 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 is produced. Symantec recommends that you must localize the ResLogLevel attribute for particular resource.

Caution: You may consider to temporarily increase the timeout values for SAPNW04 for debugging purposes. Once the debugging process is complete, you can revert back to the original timeout values.

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 Test the identified resource. The operation reproduces the problem that you are attempting to diagnose.
 - 5 Set the ResLogLevel attribute back to **INFO** for the identified resource:

```
hares -modify Resource_Name ResLogLevel INFO -sys SysA
```
 - 6 Review the contents of the VCS engine output log file.
- You may also contact Symantec support for more help.

Using trace level logging for preonline trigger

While executing the preonline trigger, you can set the [ResLogLevel](#) attribute to **TRACE**, to enable detailed logging. For more information about preonline trigger, refer to “[Configuring SAPNW04 preonline script](#)” on page 35.

To set the ResLogLevel attribute for preonline trigger

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

```
#-----  
# Define variables..  
#-----  
my $sCmd = '/opt/VRTSagents/ha/bin/SAPNW04/preonline';  
my $sResLogLevel = 'INFO'; # Define logging level..
```

```
my @lsCmdArgs = ( @ARGV, $sResLogLevel ); # Insert logging
level..
my $$Args = join ( ' ', @lsCmdArgs );
my $iExitCode = undef;
```

3 Edit the value of the ResLogLevel attribute:

```
#-----
# Define variables..
#-----
my $sCmd = '/opt/VRTSagents/ha/bin/SAPNW04/preonline';
my $sResLogLevel = 'TRACE'; # Define logging level..
my @lsCmdArgs = ( @ARGV, $sResLogLevel ); # Insert logging
level..
my $$Args = join ( ' ', @lsCmdArgs );
my $iExitCode = undef;
```

4 Save and close the preonline file.

You can view the logs in the VCS engine log,
/var/VRTSvcs/log/engine_A.log.

Sample Configurations

This chapter covers samples of agent type definition while working with a cluster. This chapter also covers sample service group configuration for SAP NetWeaver while working in a cluster.

Sample agent type definition

While working with VCS

After importing the agent types into the cluster, if you save the configuration on your system disk using the `haconf -dump` command, you can find the `SAPNW04Types.cf` file in the `/etc/VRTSvcs/conf/config/cluster` configuration directory.

An excerpt from this file follows.

```
type SAPNW04 (
    static str ArgList[] = { ResLogLevel, State, IState,
        EnqSrvResName, EnvFile, InstName, InstType,
        MonitorProgram, ProcMon, SAPAdmin, SAPMonHome,
        SAPSID, SecondLevelMonitor, StartProfile }
    str ResLogLevel = INFO
    str EnqSrvResName
    str EnvFile
    str InstName = DVEBMGS00
    str InstType = CENTRAL
    str MonitorProgram
    str ProcMon
    str SAPAdmin
    str SAPMonHome
    str SAPSID
    int SecondLevelMonitor
    str StartProfile
)
```

Sample SAP resource configuration

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

The four basic configuration include a Central instance, a Dialog instance, an Enqueue server and an Enqueue Replication server.

Sample SAP Central instance

An excerpt of the `main.cf` file for a SAP Central instance is as follows.

```
SAPNW04 SAPNW04JA_ci (  
  EnvFile = "/usr/sap/JA1/sapenv"  
  InstName = DVEBMGS05  
  InstType = CENTRAL  
  MonitorProgram = "/home/JA1/scripts/MonitorProgram.sh"  
  ProcMon = "dw se jc co"  
  ResLogLevel = INFO  
  SAPAdmin= jaladm  
  SAPMonHome = "/usr/sap/JA1/sapinfo/rfcsdk/bin"  
  SAPSID = JA1  
  SecondLevelMonitor = 1  
  StartProfile =  
  "/usr/sap/JA1/SYS/profile/START_DVEBMGS05_sapaixja"  
)
```

Sample SAP Dialog instance

An excerpt of the `main.cf` file for a SAP Dialog instance is as follows.

```
SAPNW04 SAP4NW04DI_app (  
  EnvFile = "/usr/sap/JA1/sapenv"  
  InstName = D04  
  InstType = DIALOG  
  MonitorProgram = "/home/JA1/scripts/MonitorProgram.sh"  
  ProcMon = "dw se"  
  ResLogLevel = INFO  
  SAPAdmin= jaladm  
  SAPMonHome = "/usr/sap/JA1/SYS/exe/run"  
  SAPSID = JA1  
  SecondLevelMonitor = 1  
  StartProfile = "/usr/sap/JA1/SYS/profile/START_D04_sapaixdi"  
)
```

Sample SAP ABAP Enqueue Server instance

An excerpt of the `main.cf` file for a SAP ABAP Enqueue Server instance is as follows.

```
SAPNW04 SAPNW04JA_ascs (  
  EnvFile = "/usr/sap/JA1/sapenv"
```

```
InstName = ASCS02
InstType = AENQUEUE
MonitorProgram = "/home/JA1/scripts/MonitorProgram.sh ASCS02"
ProcMon = "en ms"
ResLogLevel = INFO
SAPAdmin= jaladm
SAPMonHome = "/usr/sap/JA1/SYS/exe/run"
SAPSID = JA1
SecondLevelMonitor = 1
StartProfile = "/usr/sap/JA1/SYS/profile/START_ASCS02_sapaixja"
)
```

Sample SAP Java Enqueue Server instance

An excerpt of the `main.cf` file for a SAP Java Enqueue Server instance is as follows.

```
SAPNW04 SAPNW04JA_jscs (
  EnvFile = "/usr/sap/JA1/sapenv"
  InstName = JSCS06
  InstType = JENQUEUE
  MonitorProgram = "/home/JA1/scripts/MonitorProgram.sh JSCS06"
  ProcMon = "en ms"
  ResLogLevel = INFO
  SAPAdmin= jaladm
  SAPMonHome = "/usr/sap/JA1/SYS/exe/run"
  SAPSID = JA1
  SecondLevelMonitor = 1
  StartProfile = "/usr/sap/JA1/SYS/profile/START_JSCS06_sapaixja"
)
```

Sample SAP ABAP Enqueue Replication Server instance

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

```
SAPNW04 SAPNW04REP_arep (
  EnvFile = "/usr/sap/JA1/sapenv"
  EnqSrvResName = SAPNW04JA_ascs
  InstName = AREP02
  InstType = AENQREP
  MonitorProgram = "/home/JA1/scripts/MonitorProgram.sh"
  ProcMon = "er"
  ResLogLevel = TRACE
  SAPAdmin= jaladm
  SAPMonHome = "/usr/sap/JA1/SYS/exe/run"
  SAPSID = JA1
  SecondLevelMonitor = 1
  StartProfile =
"/usr/sap/JA1/SYS/profile/START_AREP02_sapaixrep"
)
```

Sample SAP Java Enqueue Replication Server instance

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

```
SAPNW04 SAPNW04REP_jrep (  
  EnvFile = "/usr/sap/JA1/sapenv"  
  EnqSrvResName = SAPNW04JA_jscs  
  InstName = REP06  
  InstType = JENQREP  
  MonitorProgram = "/home/JA1/scripts/MonitorProgram.sh"  
  ProcMon = "er"  
  ResLogLevel = TRACE  
  SAPAdmin= jaladm  
  SAPMonHome = "/usr/sap/JA1/SYS/exe/run"  
  SAPSID = JA1  
  SecondLevelMonitor = 1  
  StartProfile =  
  "/usr/sap/JA1/SYS/profile/START_JREP06_sapaixrep"  
)
```

Sample service group configuration for ABAP and Java architectures

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

These characteristics include:

- The SAP Central instance server or the Enqueue server must be dependent on the database server.
- Each SAP instance server to include the Central, Dialog, Enqueue, and Enqueue Replication servers should have a separate virtual IP address assigned to facilitate network transparency.
- Each SAP instance server to include the Central, Dialog, Enqueue and Enqueue Replication servers 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).

Some sample service configuration are presented below.

Figure A-1 Active-Active cluster configuration

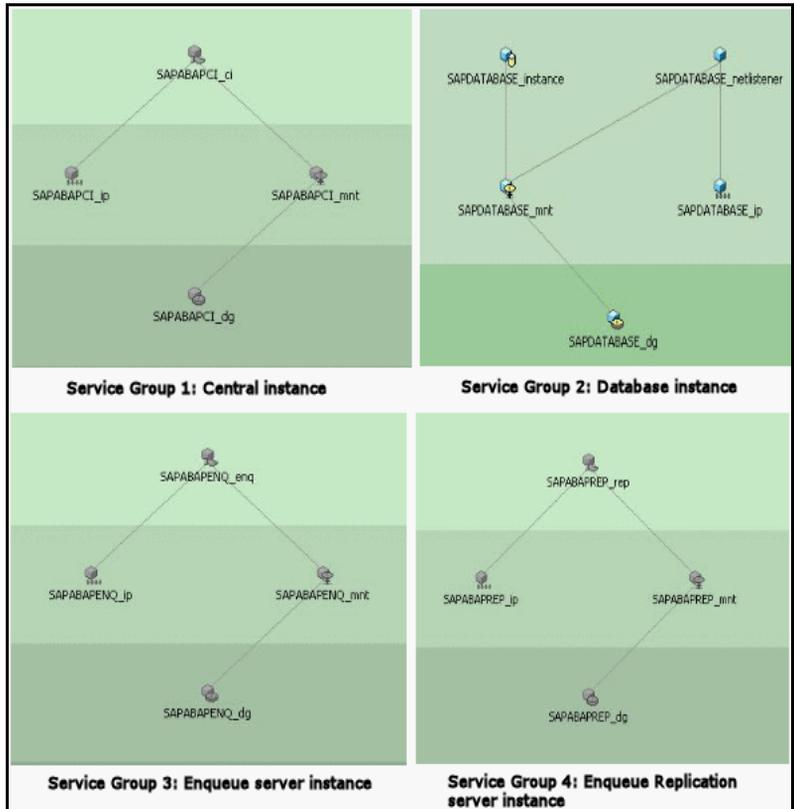


Figure A-2 Active-Passive cluster configuration

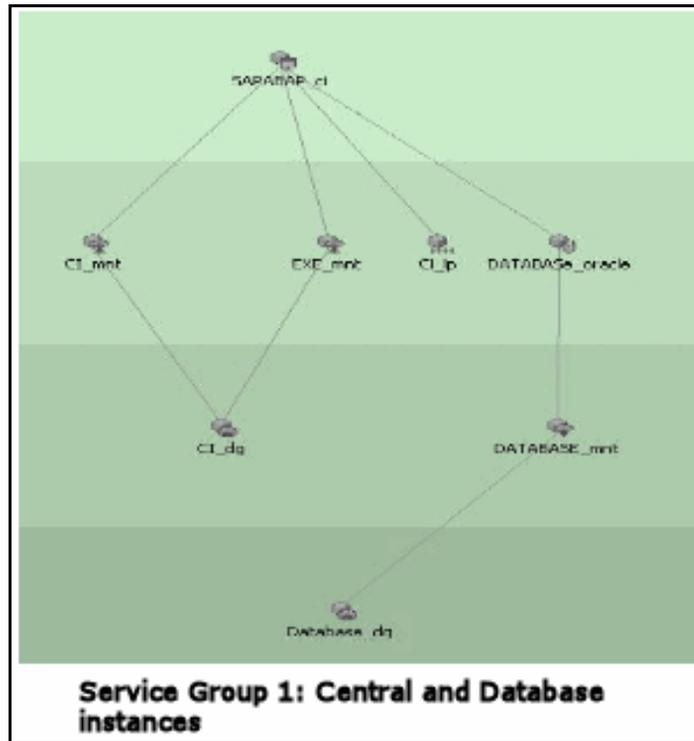
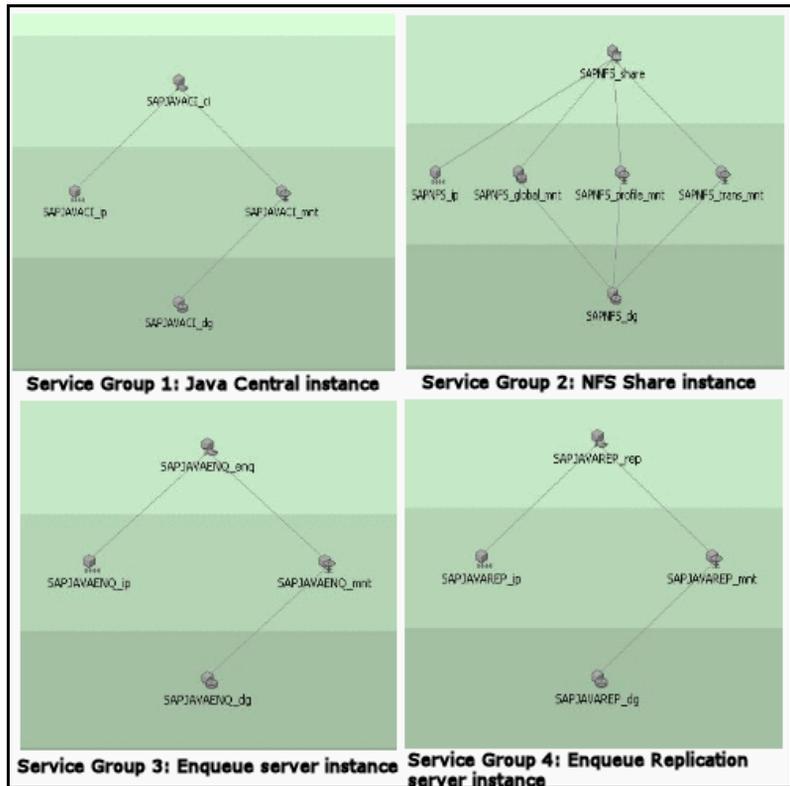


Figure A-3 Active-Active with Enqueue server configuration



Assuming that the value of the SAPSID attribute is P47, the following table covers details about how you can configure a SAP resource.

Service Group	Sample Instance Name	Sample Instance ID	Configuration details
SAP47ENQ	ASCS00	00	Mount Point: /usr/sap/P47/ASCS00 Virtual IP Host Name: ENQSRV Profile Directory: /sapmnt/P47/profile Start Profile: START_ASCS00_ENQSRV Instance Profile: P47_ASCS00_ENQSRV

Service Group	Sample Instance Name	Sample Instance ID	Configuration details
SAP47ENQREP	ENREP00	00	Mount Point: /usr/sap/P47/ENREP00 Virtual IP Host Name: ENQREP Profile Directory: /sapmnt/P47/profile Start Profile: START_ENREP00_ENQREP Instance Profile: P47_ENREP00_ENQREP
SAP47DB	NA	NA	Mount Point: /usr/database/DB_Version Virtual IP Host Name: DBSERV
SAP47DVBGS	DVBGS01	01	Mount Point: /usr/sap/P47/DVBGS01 Virtual IP Host Name: CentralHost Profile Directory: /sapmnt/P47/profile Start Profile: START_ DVBGS01_CentralHost Instance Profile: P47_ DVBGS01_CentralHost
SAP47NFSSRV	NA	NA	Mount Point: /export/sapmnt Virtual IP Host Name: SAPNFSSRV Contains directories for profile, global, and exe.
SAP47NFCNT	NA	NA	Parallel NFS Client resource that mounts SAPNFSSRV:/export/sapmnt/ P47 to /sapmnt/P47 on each node in the cluster.

Sample service group configuration for Java Add-In architecture

The common characteristics that must be part of the configuration design in this architecture are:

- The Central instance server, and Java and ABAP Enqueue server service group must be globally dependent on the database server service group.
- The Java and ABAP Enqueue servers and the Central instance must be configured in one service group.
- The Java and ABAP Replication servers must be configured in one service group.
- To facilitate network transparency:
 - Each SAP server that hosts a Central or Dialog instance must have a separate virtual IP address.
 - The Java and ABAP Enqueue Server, and the Central instance must have the same Virtual IP address.
 - The SAP Java and ABAP Enqueue Replication servers should have a same virtual IP address.
- To facilitate cluster node transparency, each SAP server that hosts a Central, Dialog, ABAP Enqueue, Java Enqueue, Java and ABAP Enqueue Replication instances must be placed on shared disk.
- Common file systems that include profile, global, and transaction file systems must be managed from one or more shared disk objects. These systems must be available to the SAP application through applications such as NFS, Cluster File System, and so on.

Some sample service group configuration is presented below.

Figure A-4 Java Add-In Standalone Enqueue Server Configuration

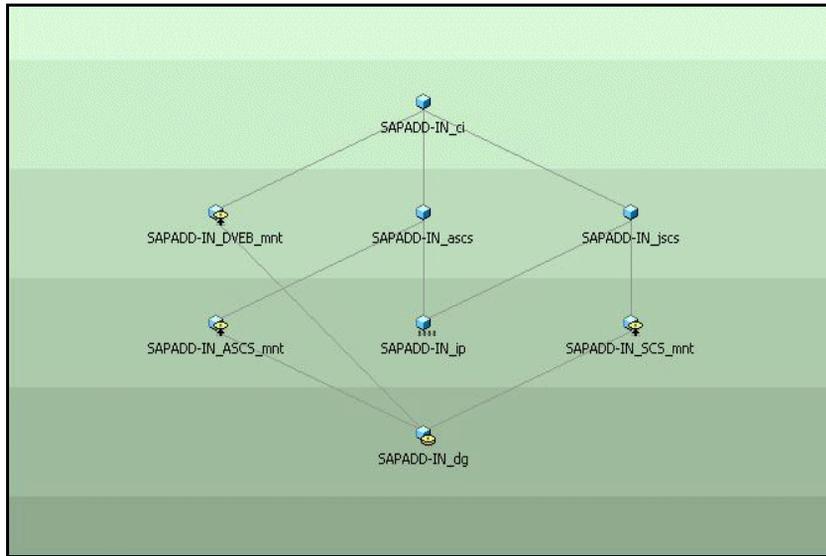
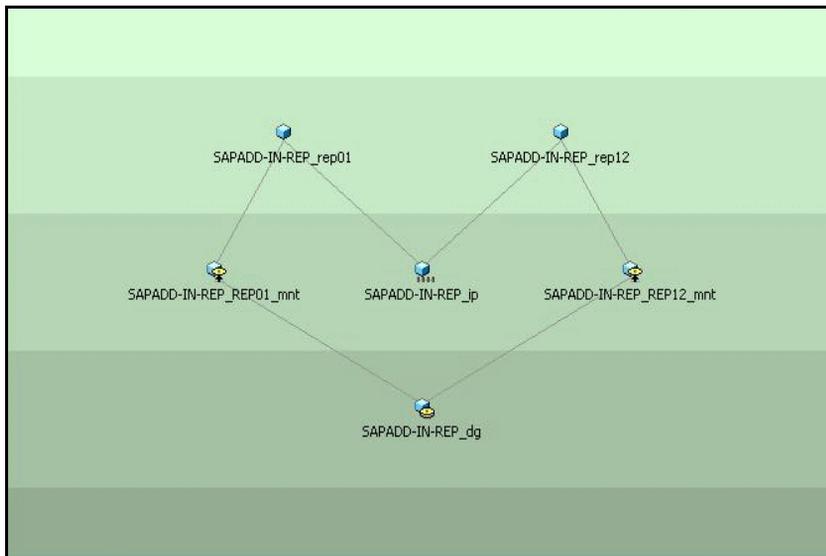


Figure A-5 Java Add-In Enqueue Replication Server Configuration



Assuming that the value of the SAPSID attribute is JA1, the following table covers details about how you can configure a SAP resource, when the SAP architecture is Java Add-In.

Service group	Resource name	Sample instance name	Configuration details
SAPNW04JA	SAPNW04J A_ci	DVEBMGS05	Mount Point: /usr/sap/JA1/DVEBMGS05 Virtual IP Host Name: SAPAIXJA Profile Directory: /sapmnt/JA1/profile Start Profile: START_DVEBMGS05_SAPAIXJA Instance Profile: JA1_DVEBMGS05_SAPAIXJA
SAPNW04JA	SAPNW04J A_ascs	ASCS02	Mount Point: /usr/sap/JA1/ASCS02 Virtual IP Host Name: SAPAIXJA Profile Directory: /sapmnt/JA1/profile Start Profile: START_ASCS02_SAPAIXJA Instance Profile: JA1_ASCS02_SAPAIXJA
SAPNW04JA	SAPNW04J A_jscs	JSCS06	Mount Point: /usr/sap/JA1/JSCS06 Virtual IP Host Name: SAPAIXJA Profile Directory: /sapmnt/JA1/profile Start Profile: START_JSCS06_SAPAIXJA Instance Profile: JA1_JSCS06_SAPAIXJA

Service group	Resource name	Sample instance name	Configuration details
SAPNW04R EP	SAPNW04R EP_arep	AREP02	<p>Mount Point: /usr/sap/JA1/AREP02</p> <p>Virtual IP Host Name: SAPAIXREP</p> <p>Profile Directory: /sapmnt/JA1/profile</p> <p>Start Profile: START_AREP02_SAPAIXREP</p> <p>Instance Profile: JA1_AREP02_SAPAIXREP</p>
SAPNW04R EP	SAPNW04R EP_jrep	JREP06	<p>Mount Point: /usr/sap/JA1/JREP06</p> <p>Virtual IP Host Name: SAPAIXREP</p> <p>Profile Directory: /sapmnt/JA1/profile</p> <p>Start Profile: START_JREP06_SAPAIXREP</p> <p>Instance Profile: JA1_JREP06_SAPAIXREP</p>
SAPNW04D B	SAPNW04D B_ora	NA	<p>Mount Point: /usr/database/DB_Version</p> <p>Virtual IP Host Name: DBSERV</p>
SAPNW04DI	SAPNW04D I_app	D04	<p>Mount Point: /usr/sap/JA1/D04</p> <p>Virtual IP Host Name: SAPAIXDI</p> <p>Profile Directory: /sapmnt/JA1/profile</p> <p>Start Profile: START_D04_SAPAIXDI</p> <p>Instance Profile: JA1_D04_SAPAIXDI</p>
SAPNW04N FSSRV	SAPNW04N FSSRV	NA	<p>Mount Point: /export/sapmnt</p> <p>Virtual IP Host Name: SAPNFSSRV</p> <p>Contains directories for profile, global, and exe.</p>

Service group	Resource name	Sample instance name	Configuration details
SAPNW04N FSCNT	SAPNW04N FSCNT	NA	Parallel NFS Client resource that mounts SAPNFSSRV:/export/sapmnt/JA1 to /sapmnt/JA1 on each node in the cluster.

Index

A

- ACC library 14
- agent
 - clustering overview 27
 - clustering SAP Servers 27
 - configuring agent 17
 - introduction 7
 - supported software 8
 - uninstalling 39
 - upgrading 14
- agent attributes 18
 - EnqSrvResName 19
 - EnvFile 19
 - InstName 19
 - InstType 19
 - MonitorProgram 21
 - ProcMon 20
 - ResLogLevel 20
 - SAPAdmin 20
 - SAPMonHome 20
 - SAPSID 20
 - SecondLevelMonitor 21
 - StartProfile 21
- agent installation on vcs 15
 - installing on aix 15
 - installing on hp-ux 15
 - installing on solaris 15
- agent operations 9
 - clean 10
 - monitor 10
 - offline 9
 - online 9
- agent uninstallation 39
 - uninstalling on aix 40
 - uninstalling on hp-ux 40
 - uninstalling on solaris 40

I

- installation prerequisites 13
 - vcs 13

P

- preonline script 35
 - configuring 35

S

- sample SAP resource configuration 50
 - Central Instance 50
 - Dialog Instance 50
 - Enqueue Replication Server Instance 51
 - Enqueue Server Instance 50
- Sample service group configuration (ABAP, Java) 52
 - active-active 53
 - active-active with enqueue server 55
 - active-passive 54
- Sample service group configuration (Java Add-In) 57
 - Enqueue Replication server configuration 58
 - Standalone Enqueue server configuration 58

