Cluster Server Agent for SAP NetWeaver Installation and Configuration Guide

AIX, Linux, Solaris

8.0

VERITAS™

Veritas Technologies LLC
2625 Augustine Drive
Santa Clara, CA 95054
http://www.veritas.com

Technical Support

Technical Support maintains support centers globally. All support services will be delivered in accordance with your support agreement and the then-current enterprise technical support policies. For information about our support offerings and how to contact Technical Support, visit our website:

https://www.veritas.com/support

You can manage your Veritas account information at the following URL:

https://my.veritas.com

If you have questions regarding an existing support agreement, please email the support agreement administration team for your region as follows:

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

Documentation

Make sure that you have the current version of the documentation. Each document displays the date of the last update on page 2. The latest documentation is available on the Veritas website:

https://sort.veritas.com/documents

Documentation feedback

Your feedback is important to us. Suggest improvements or report errors or omissions to the documentation. Include the document title, document version, chapter title, and section title of the text on which you are reporting. Send feedback to:

infoscaledocs@veritas.com

You can also see documentation information or ask a question on the Veritas community site:

http://www.veritas.com/community/

Veritas Services and Operations Readiness Tools (SORT)

Veritas Services and Operations Readiness Tools (SORT) is a website that provides information and tools to automate and simplify certain time-consuming administrative tasks. Depending on the product, SORT helps you prepare for installations and upgrades, identify risks in your datacenters, and improve operational efficiency. To see what services and tools SORT provides for your product, see the data sheet:

https://sort.veritas.com/data/support/SORT_Data_Sheet.pdf

Contents

Introducing the agent for SAP NetWeaver

This chapter includes the following topics:

- About the Cluster Server Agent for SAP NetWeaver

- Supported software

- How the agent makes SAP NetWeaver highly available

- Features of the agent

- SAP NetWeaver agent functions

- Typical SAP NetWeaver configuration in a VCS cluster

- Setting up SAP NetWeaver in a VCS cluster

About the Cluster Server Agent for SAP NetWeaver

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

The Cluster Server agent for SAP NetWeaver provides high availability for SAP NetWeaver in a cluster. The agent brings SAP instances online, monitors the instances, and takes the instances offline. The agent monitors the system processes and server states, and can shut down 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 types of SAP systems:

- ABAP

- Java

- Add-In (ABAP + Java).

The Cluster Server agent for SAP NetWeaver enables you to integrate VCS with SAP NetWeaver 7.x with SAP Kernel 7.20 DCK or later by using an SAP-provided library (`saphascriptco.so`) and a Veritas-provided cluster connector script (`sap_symc_cluster_connector`). This integration enables the SAP sapstartsrv component to communicate SAP instance status changes that are carried out by SAP clients to VCS. See "Integrating VCS and SAP NetWeaver by using the SAP library and connector script" on page 66.

Supported software

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

How the agent makes SAP NetWeaver highly available

The Cluster Server agent for SAP NetWeaver 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 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.

In addition to these levels of application monitoring, the agent for SAP NetWeaver is IMF-aware and uses asynchronous monitoring framework (AMF) kernel driver for IMF notification.

Thus, the agent ensures high availability for SAP instances.

High availability for SAP NetWeaver 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 NetWeaver is zone-aware and can monitor SAP instances running in non-global zones.

Features of the agent

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

- Support for validation of attributes that are based on 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 VCS 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.

SAP NetWeaver 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 for
  resources with the sapstartsrv InstType or StopService attribute set to 1. The
  SAP processes are removed 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.
  The agent directly starts the `sapstartsrv` service only if the InstType of the
  resource is SAPSTARTSRV. In case of resources with other InstType values,
  the agent first checks whether the `sapstartsrv` service is running. It starts the
  service only if it is not already running.

- Starts the SAP instance using the `sapcontrol` 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 the `sapcontrol` command to stop the relevant instance process.

- 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 for resources with sapstartsrv InstType or StopService attribute set to 1.

- Removes the SE and CO locks files from the `/usr/sap/SAPSID/InstName/data` directory for resources with sapstartsrv InstType or StopService attribute set to 1.

- If the SAP instance is APPSERV or ENQREP, the operation executes the `cleanipc` utility for resources with sapstartsrv InstType or StopService attribute set to 1.

- 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:

- Scans the process table to verify the SAP instance processes are running, depending on the value of the ProcMon attribute.
  See "Monitoring an SAP instance" on page 22.
  The agent also supports Intelligent Monitoring Framework (IMF) in the first level check. IMF enables intelligent resource monitoring. You can use the MonitorFreq key of the IMF attribute to specify the frequency at which the agent invokes the monitor function.

- Performs a thorough health check of the SAP instance as follows, if the value of LevelTwoMonitorFreq is greater than 0:

  - For APPSERV instances, the function uses the following utilities to perform this check:

Server type	SAP utility used
SAP NetWeaver as ABAP	sapcontrol

SAP NetWeaver as Java	sapcontrol
SAP NetWeaver as Add-In	sapcontrol

- For Enqueue Server and Enqueue Replication Server instances, the function uses the `enq_admin` or the `ensmon` utility depending on the SAP version.

- Restarts the enqueue replication server (ERS) if it is not already running.

Note: For ERS, if the RestartLimit attribute is set to 0, the agent restarts ERS at every monitor cycle.

- Executes a custom monitor utility.
  See "Executing a customized monitoring program" on page 64.

- For S/4 HANA 1809 and later:
  If the Enqueue Server 2 and the Enqueue Replication Server 2 are running on the same node, the agent checks the lock transfer state. If the lock transfer is complete, the Enqueue Replication Server 2 is switched to another available cluster node.

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 for resources with the sapstartsrv InstType or StopService attribute set to 1.

- Removes the SE and CO lock files from the `/usr/sap/SAPSID/InstName/d`ata directory for resources with the sapstartsrv InstType or StopService attribute set to 1.

- If the SAP Instance is APPSERV or ENQREP, the operation executes the `cleanipc` utility for resources with the sapstartsrv InstType or StopService attribute set to 1.

- Augments the SAP log.

imf_init

This function initializes the SAP NetWeaver agent to interface with the AMF kernel driver, which is the IMF notification module for the agent for SAP NetWeaver. This function runs when the agent starts up.

imf_getnotification

This function gets notifications about resource state changes. This function runs after the agent initializes with the AMF kernel module. This function continuously waits for notification and takes action on the resource upon notification.

imf_register

This function registers or unregisters resource entities with the AMF kernel module. This function runs for each resource after the resource goes into a steady state—online or offline.

Typical SAP NetWeaver configuration in a VCS cluster

A typical SAP NetWeaver 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 NetWeaver instance binaries are installed locally on both nodes or on shared disks.

- The Cluster Server agent for SAP NetWeaver is installed on the both nodes.

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

Figure 1-1   Typical configuration where SAP NetWeaver instance binaries
and sapmnt are installed completely on shared disks
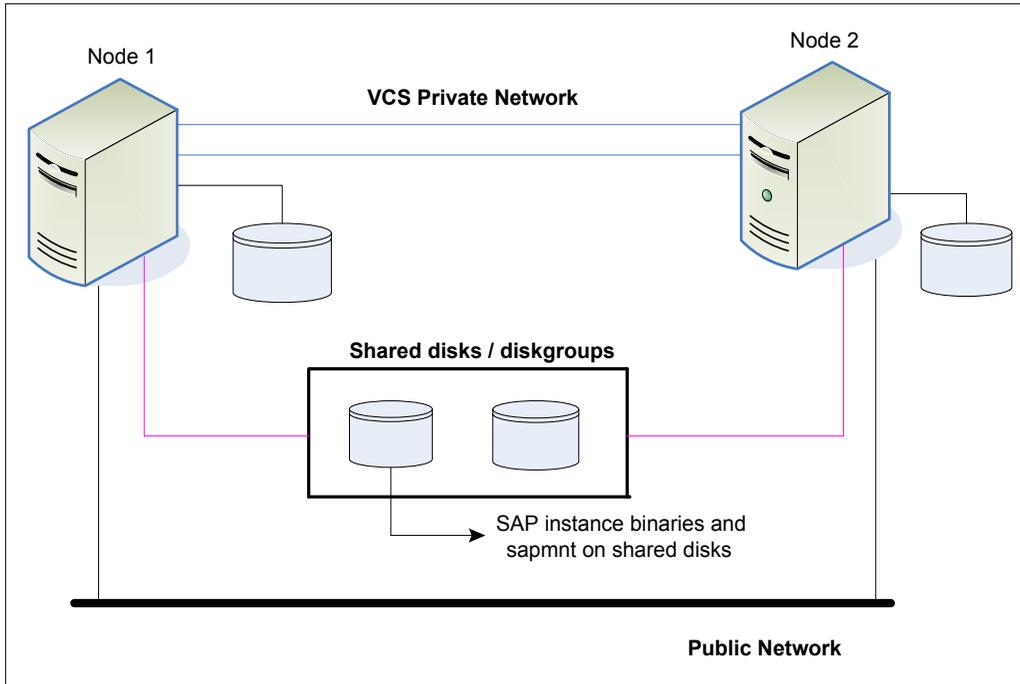
Figure 1-2 depicts a configuration where SAP NetWeaver instance binaries are
installed locally on each node and sapmnt is installed on shared disks.

Figure 1-2   Typical configuration where SAP NetWeaver instance binaries
                   are installed locally on each node and sapmnt is installed on
                   shared disks
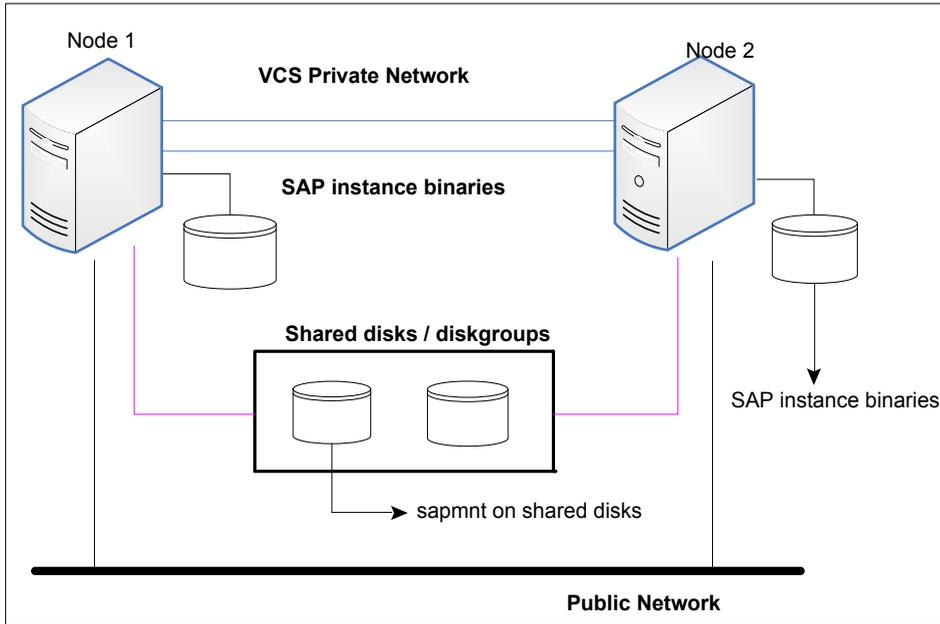
Setting up SAP NetWeaver in a VCS cluster

Perform the following tasks to set up SAP NetWeaver in a cluster:

1.   Set up a VCS cluster.

2.   Install and configure SAP NetWeaver for high availability.

3.   Install the Cluster Server agent for SAP NetWeaver.

     See "Installing the agent " on page 46.

4.   Configure the service groups for SAP NetWeaver.

     See "About configuring service groups for SAP NetWeaver" on page 73.

Installing and configuring SAP NetWeaver for high availability

This chapter includes the following topics:

- About SAP NetWeaver

- Uniquely identifying SAP NetWeaver server instances

- Monitoring an SAP instance

- About configuring SAP NetWeaver for high availability

- Configuring SAP server instances for cluster support

- Clustering shared file systems

- Configuring the SAP NetWeaver agent for message server restart

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

- Configuring CCMS Monitoring Agent for SAP instance

- Configuring the Enqueue Replication Server for SAP WebAS

About SAP NetWeaver

The SAP NetWeaver Application Server is the central foundation for the entire SAP software stack. It also provides a platform for other NetWeaver components (Portal, XI, and so on), as well as for ABAP and Java applications.

There are the following installation options for the SAP NetWeaver Application Server:

- SAP NetWeaver ABAP (ABAP only)

- SAP NetWeaver Java (Java only)

- SAP NetWeaver Add-In (ABAP and Java)

Depending on the SAP NetWeaver component to be installed, the installation option is determined. For example, SAP NetWeaver PI requires the SAP NetWeaver 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 NetWeaver 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.

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 ■ Acts as a special program in charge of some specific tasks.
	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 NetWeaver 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.

Table 2-1   SAP architecture *(continued)*

Architecture	Component	Service	Functions
SAP NetWeaver 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.
		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.

Table 2-1   SAP architecture *(continued)*

Architecture	Component	Service	Functions
	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 NetWeaver 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 NetWeaver instances in a single cluster.

For multiple instances running concurrently on a single node, the agent must be able to uniquely identify each SAP NetWeaver 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	`enq ms` for S/4 HANA 1809 and later versions `en ms` for earlier versions of SAP NetWeaver `gwrd wd` are optional
ABAP	ENQREP	`enqr` for S/4 HANA 1809 and later versions `er` for earlier versions of SAP NetWeaver
Java	APPSERV	`jc` `ig` is optional
Java	ENQUEUE	`enq ms` for S/4 HANA 1809 and later versions `en ms` for earlier versions of SAP NetWeaver `gwrd wd` are optional
Java	ENQREP	`enqr` for S/4 HANA 1809 and later versions `er` for earlier versions of SAP NetWeaver
Add-In (ABAP +Java)	APPSERV	`dw jstart` for SAP NetWeaver 7.1 or later or `dw jc` for SAP NetWeaver earlier than 7.1 `ig co se gwrd icman` are optional
Add-In (ABAP +Java)	ENQUEUE (ABAP)	`en ms`

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

SAP installation type	SAP instance type	Value of ProcMon attribute
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`
ABAP, JAVA, Add-in (ABAP +JAVA)	SAPSTARTSRV	`sapstartsrv`

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 NetWeaver for high availability

The guidelines for configuring SAP NetWeaver 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 on 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.

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

Veritas recommends maintaining 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 NetWeaver agent for message server restart

In case the message server process fails, the Cluster Server agent for SAP NetWeaver 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 NetWeaver.

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

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

For example, the 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 NetWeaver 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 LOFS, NFS fails to permit this. Due to limitations in NFS protocol it is not possible to loop back a file system, 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
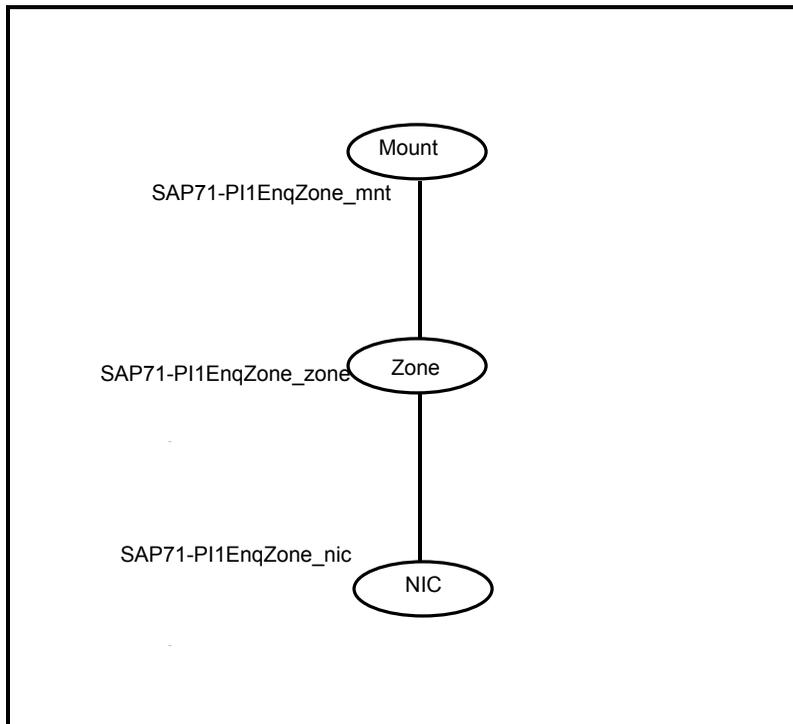

```
include "types.cf"

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 the 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 NetWeaver components
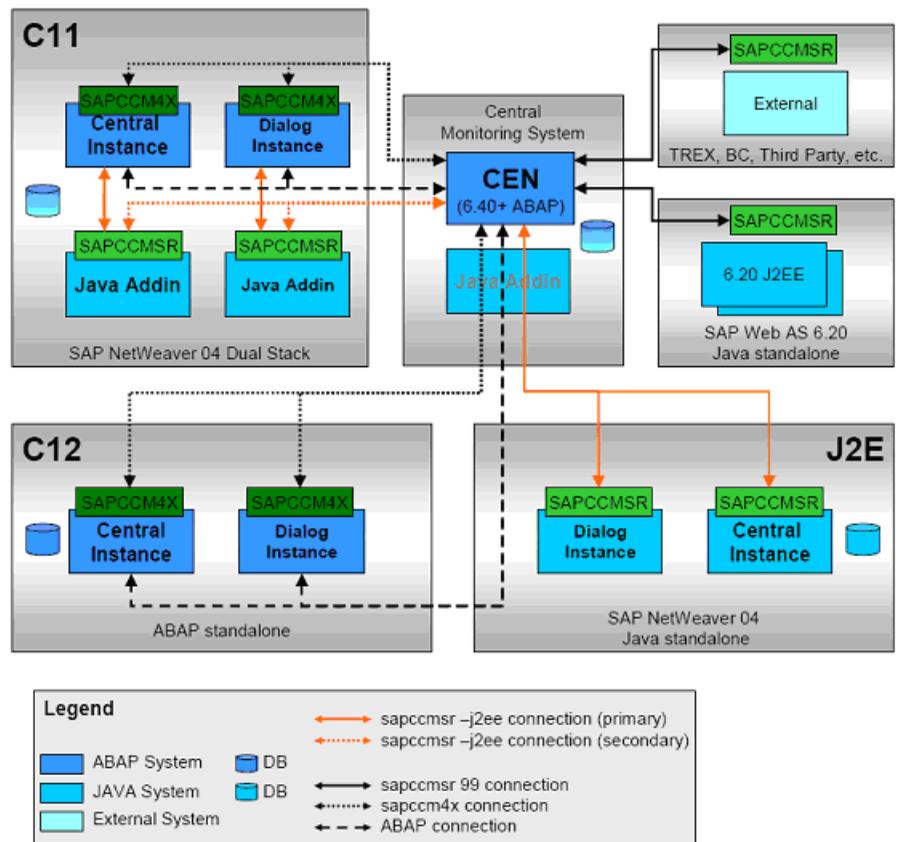
Prerequisites for installing and registering the CCMS Monitoring Agent

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 NetWeaver

The Cluster Server agent for SAP NetWeaver 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 NetWeaver.

To configure the CCMS agent with the Cluster Server agent for SAP NetWeaver

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 and removing the agent for SAP NetWeaver

This chapter includes the following topics:

- Before you install the Cluster Server agent for SAP NetWeaver
- Prerequisites for installing the agent to support Solaris zones
- Installing the ACC library
- Installing the agent
- Removing the agent
- Removing the ACC library
- Upgrading the agent for SAP NetWeaver

Before you install the Cluster Server agent for SAP NetWeaver

You must install the Cluster Server agent for SAP NetWeaver on all the systems on which you need to host a SAP NetWeaver service group.

Ensure that you meet the following prerequisites to install the agent for SAP NetWeaver:

- Install and configure Cluster Server.
  For more information, refer to the Veritas documentation that corresponds to the Cluster Server version that you plan to use.
- Install the latest version of the ACC library 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.

See "Installing the ACC library" on page 43.

Prerequisites for installing the agent to support Solaris zones

Ensure that you meet the following prerequisites to install the agent for SAP NetWeaver:

- 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.

Installing the ACC library

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

To install the ACC library

1   Log in as superuser.

2   Download ACC library 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 the individual ACCLib tar file.

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
```

Linux
```
cd1/linux/generic/vcs/application/acc_library/
version_library/rpms
```

Solaris
```
cd1/solaris/dist_arch/vcs/application/acc_library/
version_library/pkgs
```

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
```

Linux
```
# rpm -I \
VRTSacclib-VersionNumber-GA_GENERIC.noarch.rpm
```

Solaris
```
# pkgadd -d VRTSacclib.pkg
```

Installing the ACC library IPS package on Oracle Solaris 11 systems

Install the ACC library IPS package on an Oracle Solaris 11 system.

To install the ACC library IPS package on Oracle Solaris 11 systems

1 Copy the `VRTSacclib.p5p` package from the `pkgs` directory to the system in the `/tmp/install` directory.

2 Disable the publishers that are not reachable as package install may fail, if any, of the already added repositories are unreachable.

```
# pkg set-publisher --disable <publisher name>
```

3 Add a file-based repository in the system.

```
# pkg set-publisher -g /tmp/install/VRTSacclib.p5p Veritas
```

4 Install the package.

```
# pkg install --accept VRTSacclib
```

5 Remove the publisher from the system.

```
# pkg unset-publisher Veritas
```

6 Enable the publishers that were disabled earlier.

```
# pkg set-publisher --enable <publisher name>
```

Installing the ACC library package on Solaris brand non-global zones

With Oracle Solaris 11, you must install the ACC library package inside non-global zones. The native non-global zones are called Solaris brand zones.

To install the ACC library package on Solaris brand non-global zones

1 Ensure that the SMF services,
`svc:/application/pkg/system-repository:default` and
`svc:/application/pkg/zones-proxyd:default`, are online on the global zone.

```
# svcs svc:/application/pkg/system-repository:default
```

```
# svcs svc:/application/pkg/zones-proxyd:default
```

2 Log on to the non-global zone as a superuser.

3 Ensure that the SMF service
`svc:/application/pkg/zones-proxy-client:default` is online inside the non-global zone:

```
# svcs svc:/application/pkg/zones-proxy-client:default
```

4 Copy the `VRTSacclib.p5p` package from the `pkgs` directory to the non-global zone (for example, at the `/tmp/install` directory).

5 Disable the publishers that are not reachable, as package install may fail, if any of the already added repositories are unreachable.

```
# pkg set-publisher --disable <publisher name>
```

6 Add a file-based repository in the non-global zone.

```
# pkg set-publisher -g/tmp/install/VRTSacclib.p5p Veritas
```

7 Install the package.

```
# pkg install --accept VRTSacclib
```

8 Remove the publisher on the non-global zone.

```
# pkg unset-publisher Veritas
```

9 Clear the state of the SMF service, as setting the file-based repository causes the SMF service `svc:/application/pkg/system-repository:default` to go into the maintenance state.

```
# svcadm clear svc:/application/pkg/system-repository:default
```

10 Enable the publishers that were disabled earlier.

```
# pkg set-publisher --enable <publisher>
```

Note: Perform steps 2 through 10 on each non-global zone.

Installing the agent

Install the agent for SAP NetWeaver on each node in the cluster.

To install the agent 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`
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`
	
	dist_arch is sol_sparc.

4 Log in as superuser.

5   Install the package.

AIX   # installp -ac -d VRTSsapnw.rte.bff VRTSsapnw.rte

Linux   # rpm -ihv \
                   VRTSsapnw-*AgentVersion*-GA_GENERIC.noarch.rpm

Solaris   # pkgadd -d . VRTSsapnw

6   After installing the agent package, you must import the agent type configuration
file.

See "Importing the agent types files in a VCS environment" on page 54.

Installing the agent IPS package on Oracle Solaris 11 systems

To install the agent IPS package on an Oracle Solaris 11 system

1   Copy the VRTSsapnw.p5p package from the pkgs directory to the system in the
/tmp/install directory.

2   Disable the publishers that are not reachable as package install may fail, if any
of the already added repositories are unreachable.

pkg set-publisher --disable <*publisher name*>

where the publisher name is obtained using the pkg publisher command.

3   Add a file-based repository in the system.

pkg set-publisher -g /tmp/install/VRTSsapnw.p5p Veritas

4   Install the package.

pkg install --accept VRTSsapnw

5   Remove the publisher from the system.

pkg unset-publisher Veritas

6   Enable the publishers that were disabled earlier.

pkg set-publisher --enable <*publisher name*>

Installing agent packages on Solaris brand non-global zones

To install the agent package on Solaris brand non-global zones

1 Ensure that the SMF services,
`svc:/application/pkg/system-repository:default` and
`svc:/application/pkg/zones-proxyd:default`, are online on the global
zone.

```
# svcs svc:/application/pkg/system-repository:default
```

```
# svcs svc:/application/pkg/zones-proxyd:default
```

2 Log on to the non-global zone as a superuser.

3 Ensure that the SMF service
`svc:/application/pkg/zones-proxy-client:default` is online inside
non-global zone:

```
# svcs svc:/application/pkg/zones-proxy-client:default
```

4 Copy the `VRTSsapnw.p5p` package from the `pkgs` directory to the non-global
zone (for example, at the `/tmp/install` directory).

5 Disable the publishers that are not reachable, as package install may fail, if
any of the already added repositories are unreachable.

```
# pkg set-publisher --disable <publisher name>
```

6 Add a file-based repository in the non-global zone.

```
# pkg set-publisher -g/tmp/install/VRTSsapnw.p5p Veritas
```

7 Install the package.

```
# pkg install --accept VRTSsapnw
```

8 Remove the publisher on the non-global zone.

```
# pkg unset-publisher Veritas
```

9 Clear the state of the SMF service, as setting the file-based repository causes
the SMF service `svc:/application/pkg/system-repository:default` to go
into the maintenance state.

```
# svcadm clear svc:/application/pkg/system-repository:default
```

10 Enable the publishers that were disabled earlier.

```
# pkg set-publisher --enable <publisher>
```

Note: Perform steps 2 through 10 on each non-global zone.

Installing the agent in a Solaris 10 brand zone

To install the SAP NetWeaver agent in a brand zone on Solaris 10:

- Ensure that the ACClibrary 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 . VRTSsapnw
  ```
  For example: `# pkgadd -R /zones/zone1/root -d . VRTSsapnw`

Removing the agent

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

   ```
   # hares -list Type=SAPNW
   ```

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

   ```
   # hatype -delete SAPNW
   ```

   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 NetWeaver from each node in the cluster.

Execute the command pertaining to your platform to uninstall the agent:

AIX	`# installp -u VRTSsapnw.rte`
Linux	`# rpm -e VRTSsapnw`
Solaris	`# pkgrm VRTSsapnw`

Note: To uninstall the agent IPS package on a Solaris 11 system:

```
# pkg uninstall VRTSsapnw
```

For Solaris 11, uninstalling the SAP NetWeaver agent will also uninstall the SAP liveCache agent.

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 Execute the command pertaining to your platform to remove the ACC library package:

AIX	`# installp -u VRTSacclib`
Linux	`# rpm -e VRTSacclib`
Solaris	`# pkgrm VRTSacclib`

Note: To uninstall the ACC Library IPS package on a Solaris 11 system:

```
# pkg uninstall VRTSacclib
```

Upgrading the agent for SAP NetWeaver

Perform the following steps to upgrade the agent with minimal disruption.

Note: Do not use the `rpm -Uvh` command to perform the upgrade; it is not supported.

To upgrade the agent for SAP NetWeaver

1   Verify the version of the agent for SAP NetWeaver.

Linux   `# rpm -qi VRTSsapnw | grep Version`

Solaris `# pkginfo -l VRTSsapnw | grep VERSION`

The output resembles:

`Version : 7.0.6.0`

2   Save the VCS configuration.

`# haconf -dump -makero`

3   Identify the SAP NetWeaver resource and service group.

`# hatype -resources SAPNW`

The output resembles:

`sapnw_res`

`# hares -display sapnw_res | grep Group`

The output resembles:

`sapnw_res      Group      global      sapnw_res_grp`

4   Check the type-level attribute—Version.

`# hatype -display SAPNW | grep Version`

5   Freeze the SAP NetWeaver service group.

`# hagrp -freeze sapnw_grp`

6   Check if the SAP NetWeaver agent is running.

`# haagent -display SAPNW | grep Running`

The output resembles:

`SAPNW        Running        Yes`

7 If the agent is running, stop the agent.

```
# haagent -stop SAPNW -force -sys hostname
```

8 Verify the status of the agent.

```
# haagent -display SAPNW | grep Running
```

The output resembles:

```
SAPNW       Running       No
```

9 Uninstall the agent.

See "Removing the agent" on page 49.

10 Install the latest agent.

See "Installing the agent " on page 46.

11 Update the agent type definition.

```
# haconf -makerw

# hatype -modify SAPNW ArgList ResLogLevel State IState EnvFile
SAPAdmin InstProfile InstType ProcMon EnqSrvResName MonitorProgram
StopService UseSystemD

# haattr -add SAPNW UseSystemD -boolean 0

# haconf -dump -makero
```

12 Start the SAP NetWeaver agent.

```
# haagent -start SAPNW -sys hostname
```

The output resembles:

```
VCS NOTICE V-16-1-10001 Please look for messages in the log file
```

13 Verify the status of the agent.

```
# haagent -display SAPNW | grep Running
```

The output resembles:

```
SAPNW       Running       Yes
```

14 Unfreeze the SAP NetWeaver service group.

```
# hagrp -unfreeze sapnw_grp
```

4

Configuring the agent for SAP NetWeaver

This chapter includes the following topics:

About configuring the Cluster Server agent for SAP NetWeaver

After installing the Cluster Server agent for SAP NetWeaver, 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 NetWeaver resources.

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

Importing the agent types files in a VCS environment

To use the agent for SAP NetWeaver, you must import the agent types file into the cluster.

You can import the agent types file using the Cluster Manager (Java Console) or using the command line interface (CLI).

To import the agent types file using Cluster Manager (Java Console)

1 Start the Cluster Manager 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 6.0 or later	■ AIX ■ Linux	/etc/VRTSagents/ha/conf/SAPNW/ SAPNWTypes.cf
VCS 6.0 or later	Solaris	/etc/VRTSagents/ha/conf/SAPNW/ SAPNWTypes51.cf

4 Click **Import**.

5 Save the VCS configuration.

The SAP agent types file is now imported to the VCS engine.

You can now create SAP NetWeaver resources. For additional information about using the Cluster Manager (Java Console), refer to the *Cluster Server Administrator's Guide*.

To import the agent types file using the 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 6.0 or later	■ AIX	`/etc/VRTSagents/ha/conf/SAPNW/`
	■ Linux	`SAPNWTypes.cf`
VCS 6.0 or later	Solaris	`/etc/VRTSagents/ha/conf/SAPNW/`
		`SAPNWTypes51.cf`

4 Create a dummy `main.cf` file:

```
# echo 'include "SAPNWTypes.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 NetWeaver resources.

SAP NetWeaver agent attributes

Table 4-1 Required attributes for configuring an SAP NetWeaver instance

Required attributes	Description
EnqSrvResName	The name of the VCS resource for SAP Central Services (A)SCS Instance. This attribute is used by Enqueue server 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: No default value
	Example: SAP71-PI1SCS_sap

Table 4-1 Required attributes for configuring an SAP NetWeaver instance
(continued)

Required attributes	Description
EnvFile	The absolute path to the file that must be sourced with the UNIX shell. The supported shell environments are ksh, sh, and csh. The agent sources this file to set the environment before executing SAP scripts for online, offline, monitor, and clean operations.
	Note: On SystemD-enabled systems, you must specify a SystemD-compliant environment file in the SystemDEnvFile attribute.
	See "Generating an environment file for SAP" on page 85.
	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: No default value
	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
	■ ENQREP: Enqueue Replication Server
	■ SMDAGENT: Solution Manager Diagnostics Agent
	■ SAPSTARTSRV: SAPSTARTSRV Process
	Note: The value of this attribute is not case-sensitive.
	Type and dimension: string-scalar
	Default: APPSERV
	Example: ENQUEUE
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 22.
	Type and dimension: string-scalar
	Default: No default value
	Example: dw se jstart

Table 4-1   Required attributes for configuring an SAP NetWeaver instance
                     (continued)

Required attributes	Description
SAPAdmin	The 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: No default value
	Example: pi1adm
InstProfile	The full path to the SAP Instance profile.
	The *SAPSID* is found in the `/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: No default value
	Example: `/usr/sap/PI1/SYS/profile/PI1_DVEBMGS00_sappi1pas`

Table 4-2   Optional attributes for configuring an SAP NetWeaver instance

Optional attribute	Description
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 **Note:** The use of the ResLogLevel attribute is deprecated from VCS version 6.2 onwards. You must use the LogDbg attribute instead of the ResLogLevel attribute to enable debug logs for the ACCLib-based agents, when the ACCLib version is 6.2.0.0 or later. The agent captures the first failure data of the unexpected events and automatically logs debug messages in their respective agent log files.
LogDbg	For ACCLib-based agents, you must use the LogDbg resource type attribute to enable the debug logs when the ACCLib version is 6.2.0.0 or later and the VCS version is 6.2 or later. Set the LogDbg attribute to DBG_5 to enable debug logs for ACCLIB based agent. By default, setting the LogDbg attribute to DBG_5 enables debug logs for all SAPNW resources in the cluster. If debug logs must be enabled for a specific SAPNW resource, override the LogDbg attribute. Type and dimension: string-keylist Default: {} (none) For more information on how to use the LogDbg attribute, refer to the *Cluster Server Administrator's Guide*.

Table 4-2   Optional attributes for configuring an SAP NetWeaver instance
                    (continued)

Optional attribute	Description

MonitorProgram	The absolute path name of an external, user-supplied monitor executable.
	
	■ See "Executing a customized monitoring program" 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`

LevelTwoMonitorFreq	This type-level attribute specifies the frequency at which the agent for this resource type must perform the second-level or detailed monitoring. You can also override the value of this attribute at the resource level. The value indicates the number of monitor cycles after which the agent will monitor the SAP instance in detail.
	
	For example, the value 5 indicates that the agent will monitor the SAP instance in detail after every five online monitor intervals.
	
	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: `enq_admin` or `ensmon`, depending on the SAP version
	
	Type and dimension: integer-scalar
	
	Default: 0
	
	Example:
	`# /opt/VRTSvcs/bin/hatype -modify SAPNW`
	`LevelTwoMonitorFreq 1`

Table 4-2   Optional attributes for configuring an SAP NetWeaver instance
(continued)

Optional attribute	Description
IMF	This type-level attribute determines if the agent must perform intelligent resource monitoring. You can also override the value of this attribute at the resource level. See "About the keys of the IMF attribute" on page 63.
IMFRegList	An ordered list of attributes whose values are registered with the IMF notification module. The attribute values can be overridden at the resource level.
StopService	Specifies if the agent must stop SAP Start Service (sapstartsrv) for this instance. Valid values are: ■ 1 - Stops SAP Start Service for this instance. ■ 0 - Does not stop SAP Start Service for this instance. **Note:** The StopService attribute must be set to 0 for the agent to be able to use SAP HA Library with SAP NetWeaver 7.x with SAP Kernel 7.20 DCK or later. Type and dimension: boolean-scalar Default: 0 Example: 1

Table 4-2   Optional attributes for configuring an SAP NetWeaver instance
                       (continued)

Optional attribute	Description
ContainerOpts	Used in operating system virtualization technology environments. Indicates options that may be passed to the agent framework so that Container behavior can be controlled at the resource level. This attribute cannot be modified, if resources of this type are in use, unless you override for the specific resource you want modified. If overridden, can only be modified if resource is in a clean OFFLINE state. A Mount resource type does not have this attribute. In order for a Mount resource to be brought up inside a Container, you must override this attribute at the resource level and set RUNINCONTAINER to 1. ContainerOpts has the following keys: ■ RUNINCONTAINER: Valid values are 0 and 1. Assign the value to 1, if you want the resource to be managed inside a Container. Group-level attribute ContainerInfo must be set in order for this value to be effective. ■ PASSCINFO: Valid values are 0 and 1. If set to 1, the agent framework sends the Container information to the resource (agent entry points). Group-level attribute must be set in order for this value to be effective. ■ PASSLOADINFO: Valid values are 0 and 1. If set to 1, the service group's Load values are passed on to the resource verbatim. See the *Cluster Server Administrator's Guide* and the respective application virtualization guide.

Table 4-2   Optional attributes for configuring an SAP NetWeaver instance
                     (continued)

Optional attribute	Description

ContainerInfo	If this attribute is populated for a service group, VCS manages resources of that service group inside a Container depending upon the ContainerOpts attribute specified for those resource types. It is an association attribute with the following three keys:
	
	■ Name: Name is a free-form text.
	■ Type: Supported types are Zone on Solaris and WPAR on AIX.
	■ Enabled: Enabled can be:
	  ■ 0: Manages the Container resource, but manage all other resource outside of the Container.
	  ■ 1: Manages resources inside the Container based on ContainerOpts value.
	  ■ 2: No-op for the Container resource and manages all resources outside the Container.
	
	Default value: "" (no container).

UseSystemD	SystemD is a system and a service manager for Linux operating systems. It helps manage applications across Linux distributions that support the SystemD feature. When the UseSystemD attribute of the SAP NetWeaver agent is set to 1, on SLES 12 or RHEL 7 distributions, the SAP NetWeaver agent starts the SAP processes in system.slice. When this attribute is set to 0, a typical online function starts the SAP processes in user.slice. **Note:** When this attribute is set to 1, you may encounter unexpected behavior while performing the online and offline operations on SAP resources. This behavior may be a result of an improper configuration in the SAP environment file. **Note:** If SystemD is enabled, a SystemD-compliant environment file must be specified in the SystemDEnvFile attribute. This file is in addition to the environment file that is used to set the environment before the agent executes any scripts for the application that it monitors. Type and dimension: boolean-scalar Example: 1 Default: 0

Table 4-2 Optional attributes for configuring an SAP NetWeaver instance *(continued)*

Optional attribute	Description
SystemDEnvFile	Full path of the SystemD-compliant environment file. On SystemD-enabled platforms, the value of the EnvironmentFile directive is set to this value. This file is in addition to the environment file that is used to set the environment before the agent executes any scripts for the application that it monitors. See "Generating an environment file for SAP" on page 85. **Note:** You must specify a SystemD-compliant environment file in this attribute if the SystemD attribute is enabled. If you do not specify a value, the agent uses the value of the EnvFile attribute in the service file. Type and dimension: string-scalar Example: `/home/tstadm/login.env` Default: No default value

About the keys of the IMF attribute

The IMF type-level attribute uses the following keys:

Table 4-3 IMF attribute keys

Key	Description
Mode	Define this attribute to enable or disable intelligent resource monitoring. Valid values are as follows: ■ 0—Does not perform intelligent resource monitoring ■ 1—Performs intelligent resource monitoring for offline resources and performs poll-based monitoring for online resources ■ 2—Performs intelligent resource monitoring for online resources and performs poll-based monitoring for offline resources ■ 3—Performs intelligent resource monitoring for both online and for offline resources. **Note:** The agent for SAP NetWeaver supports intelligent resource monitoring for online resources only. Hence, Mode should be set to either 0 or 2. Default: 2

Table 4-3   IMF attribute keys *(continued)*

Key	Description
MonitorFreq	This key value specifies the frequency at which the agent invokes the monitor agent function. The value of this key is an integer.
	You can set this key to a non-zero value for cases where the agent requires to perform both poll-based and intelligent resource monitoring.
	If the value is 0, the agent does not perform poll-based process check monitoring.
	After the resource registers with the AMF kernel driver, the agent calls the monitor agent function as follows:
	■ After every (MonitorFreq x MonitorInterval) number of seconds for online resources ■ After every (MonitorFreq x OfflineMonitorInterval) number of seconds for offline resources
	Default: 5
RegisterRetryLimit	If you enable intelligent resource monitoring, the agent invokes the imf_register agent function to register the resource with the AMF kernel driver.
	The value of the RegisterRetryLimit key determines the number of times the agent must retry registration for a resource. If the agent cannot register the resource within the limit that is specified, then intelligent monitoring is disabled until the resource state changes or the value of the Mode key changes.
	Default: 3

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 NetWeaver instance is online.

■ The LevelTwoMonitorFreq attribute is either set to 0 or 1, and the second-level check indicates that the SAP NetWeaver instance is online.

- The LevelTwoMonitorFreq 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.

Configuring the sapstartsrv process for high availability

This section is applicable to SAP NetWeaver 7.x with SAP Kernel 7.20 DCK or later.

Before you configure VCS SAPNW resources, you must configure SAP Start Service (sapstartsrv) for high availability.

Sapstartsrv provides basic management services, such as start, stop, and monitor, for assigned SAP instances. Sapstartsrv is a vital part of each instance of the SAP system and controls the processes that run within the instance.

In a high availability setup, sapstartsrv must fail over together with the assigned SAP instance. For example, when a VCS SAPNW resource for an SAP instance fails over to another node, sapstartsrv must also fail over simultaneously. However, when an SAP instance is stopped during the offline operation of a VCS SAPNW resource, sapstartsrv must not be stopped.

On each cluster node, you must run the following commands as a root user:

```
# chmod 555 /opt/VRTSagents/ha/bin/SAPNW/sap_symc_cluster_connector
```

To achieve this behavior, you must configure the SAPNW resource as follows:

```
SAPNW PRD_PAS_SAPNW_SAPSTARTSRV (
EnvFile = "/home/prdadm/.login"
SAPAdmin = prdadm
InstProfile = "/usr/sap/PRD/SYS/profile/PRD_J01_sapprdapp"
InstType = SAPSTARTSRV
```

```
ProcMon = sapstartsrv
)
```

You must also set the RestartLimit attribute for each SAPNW resource configured
for the sapstartsrv process as shown in the following example:

```
# /opt/VRTSvcs/bin/hares -override SAP_ASCS_sapstartsrv RestartLimit
/opt/VRTSvcs/bin/hares -modify SAP_ASCS_sapstartsrv RestartLimit 3
```

You can override the attribute values as shown in the following example for the
ToleranceLimit attribute:

```
# /opt/VRTSvcs/bin/hares -override SAP_PAS_Application ToleranceLimit
/opt/VRTSvcs/bin/hares -modify SAP_PAS_Application ToleranceLimit 3
```

For resource dependency, See "Sample SAP resource configuration" on page 101.

Integrating VCS and SAP NetWeaver by using the SAP library and connector script

This section is applicable to SAP NetWeaver 7.x with SAP Kernel 7.20 DCK or
later.

In a typical VCS environment, when an SAP administrator changes the status of
an SAP instance by means of an SAP client, such as sapcontrol or startsap, then
VCS performs the following actions:

- When the administrator stops the SAP instance, VCS detects a fault and performs
  the clean operation.

- When the administrator starts the SAP instance, VCS detects that the instance
  is brought online outside of VCS control.

To ensure that VCS detects the correct status of SAP instances when the status
of instances are intentionally changed outside of VCS control, sapstartsrv must be
able to communicate with VCS. When communication between sapstartsrv and
VCS is enabled, sapstartsrv can inform VCS when an SAP client is used to start
or stop an assigned SAP instance.

The following components are involved in enabling communication between
sapstartsrv and VCS:

- SAP clients, such as sapcontrol, startsap, SAP Management Console, and so
  on. All these clients must use the web-service interface of sapstartsrv.

- The sapstartsrv server process that is responsible for starting, stopping, and
  retrieving information from the SAP instance

- The SAP-provided library saphascriptco.so

- The Veritas-provided cluster connector script sap_symc_cluster_connector

- Cluster Server

Note: Communication between VCS and sapstartsrv can be enabled when the SAP Kernel version is 7.20 or later with patch level 100 and the new directory structure of SAP Kernel 7.20 or later is used.

For SAP NetWeaver 7.40 with SAP Kernel 7.42 or later:

- A new version of integration scenario is available for integrating VCS and SAP NetWeaver by using the SAP library and connector script, sap_symc_cluster_connector. This integration scenario provides a few new API commands and enhanced test cases. The integration scenario NW-HA-CLU 7.40 is downward compatible with the previous NW-HA-CLU 7.30 release. For more information, refer to the relevant SAP NetWeaver High Availability Cluster 740 Certification documentation and *SAP Note 1693245 - SAP HA Script Connector Library*.

- Ensure that patch level 414 or later is installed.

Note: Before you upgrade a SAP kernel using the Rolling Kernel Switch (RKS) method, you must freeze all the VCS service groups that correspond to SAP application servers.

After the upgrade is complete, make sure to unfreeze the service groups.

Enabling the <sid>adm user to run VCS commands in read-write mode

Ensure that the read and execute permissions are set for the /opt/VRTS/bin directory:

```
chmod +rx /opt/VRTS/bin/
```

In Solaris, to call the intentional offline script (sap_symc_cluster_script), add "setenv VCS_HOST *<global hostname>*" in the environment file that is specified in the EnvFile attribute of the resource of that instance.

To be able to run VCS commands in read-write mode, the <sid>adm user must run the following halogin command from each node in the cluster.

```
/opt/VRTS/bin/halogin admin <PASSWORD>
```

For more information on the halogin command, see the HALOGIN (1M) man page.

Configuring sapstartsrv and VCS with saphascriptco.so

Set the following parameters in each SAP instance profile that is read by sapstartrsv.

For Linux and Solaris:

```
service/halib = $(DIR_CT_RUN)/saphascriptco.so
```

```
service/halib_cluster_connector =
/opt/VRTSagents/ha/bin/SAPNW/sap_symc_cluster_connector
```

For AIX:

```
service/halib = $(DIR_CT_RUN)/saphascriptco.o
```

```
service/halib_cluster_connector =
/opt/VRTSagents/ha/bin/SAPNW/sap_symc_cluster_connector
```

Note: Restart the sapstartsrv process for each SAP instance after modifying its profile.

Commands implemented by the sap_symc_cluster_connector script

The sap_symc_cluster_connector script is the interface between the SAP high availability cluster library and VCS. Currently, it provides commands to serve cluster requests, such as bringing the VCS resources online and taking the resources offline, from the sapstartsrv process.

The sap_symc_cluster_connector script implements the following commands.

`help` or `--help`	Prints usage of the sap_symc_cluster_connector script.
`init`	Checks if VCS is running.
`cpa --res SAPRESID --act ACT`	Checks if VCS resource for the corresponding SAP instance is in the pending ONLINE or OFFLINE state in the cluster.

`fra --res SAPRESID --act ACT [ --node NODE ]`	Fires a cluster action as listed below: ■ Brings the VCS resource online for the corresponding SAP instance, if ACT is START. ■ Takes the VCS resource offline for the corresponding SAP instance, if ACT is STOP. ■ Switches the VCS service group for the corresponding SAP instance, if ACT is MIGRATE.
`gvi --out FILE`	Prints the VCS version. This command is required since the sap script connector API version 2.
`hcc --out FILE --sid SID --ino INSTNO`	Performs high availability configuration checks, such as verifying if VCS resource for the corresponding SAP instance is defined. This command is required since the sap script connector API version 2.
`lsn --out FILE --res SAPRESID`	Lists VCS nodes for the VCS resource for the corresponding SAP instance. This command is used since the sap script connector API version 2.
`lsr --out FILE --sid SID --ino INSTNO`	Lists VCS resource and VCS service group for the corresponding SAP instance.

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 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.

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.

Enabling the agent to support IMF

In order to enable the SAP NetWeaver agent to support IMF, you must make the following configuration changes to the attributes of the agent:

- AgentFile: Set the AgentFile attribute to **Script51Agent**.

- IMF Mode: Set the IMF Mode attribute to **2**.

- IMFRegList: Update the IMFRegList attribute.

The following sections provide more information about the commands that you can use to make these configuration changes, depending on whether VCS is in a running state or not.

If VCS is in a running state

To enable the SAP NetWeaver resource for IMF when VCS is in a running state:

1 Make the VCS configuration writable.

```
# haconf -makerw
```

2 Run the following command to update the AgentFile attribute.

```
# hatype -modify SAPNW AgentFile\
/opt/VRTSvcs/bin/Script51Agent
```

3 Run the following commands to add the IMF attributes:

```
# haattr -add -static  SAPNW IMF -integer -assoc Mode 0 \
MonitorFreq 1 RegisterRetryLimit 3

# haattr -add -static SAPNW IMFRegList -string -vector
```

4 Run the following command to update the IMF attribute.

```
# hatype -modify SAPNW IMF Mode num MonitorFreq num
RegisterRetryLimit num
```

For example, to enable intelligent monitoring of online resources, with the MonitorFreq key set to 5, and the RegisterRetryLimit key is set to 3, run the following command:

```
# hatype -modify SAPNW IMF Mode 2 MonitorFreq 5 \
RegisterRetryLimit 3
```

Note: The valid values for the Mode key of the IMF attribute are 0 (disabled) and 2 (online monitoring).

5 Run the following command to update the IMFRegList attribute:

```
# hatype -modify SAPNW IMFRegList ProcMon InstProfile
```

6 Save the VCS configuration.

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

7 If the SAP NetWeaver agent is running, restart the agent.

For information on the commands you can use to restart the agent, see Restarting the agent.

Restarting the agent

To restart the agent:

1 Run the following command to stop the agent forcefully:

```
# haagent -stop <resourceType> -force -sys <systemName>
```

Note: Stopping the agent forcefully eliminates the need to take the resource offline.

2 Run the following command to start the agent:

```
# haagent -start <resourceType> -sys <systemName>
```

If VCS is not in a running state

To change the SAPNW type definition file when VCS is not in a running state:

1 Update the AgentFile attribute.

```
static str AgentFile = "/opt/VRTSvcs/bin/Script51Agent"
```

2 Update the IMF attribute.

The valid values for the Mode key of the IMF attribute are 0 (disabled) and 2 (online monitoring).

```
static int IMF{} = { Mode=num, MonitorFreq=num,
RegisterRetryLimit=num }
```

For example, to update the IMF attribute such that the Mode key is set to 2, the MonitorFreq key is set to 5, and the RegisterRetryLimit key is set to 3:

```
static int IMF{} = { Mode=2, MonitorFreq=5, RegisterRetryLimit=3
}
```

3 Update the IMFRegList attribute.

```
static str IMFRegList[] = { ProcMon, InstProfile }
```

Disabling intelligent resource monitoring

To disable intelligent resource monitoring

1 Make the VCS configuration writable.

```
# haconf -makerw
```

2 To disable intelligent resource monitoring for all the resources of a certain type, run the following command:

```
# hatype -modify SAPNW IMF -update Mode 0
```

3 To disable intelligent resource monitoring for a specific resource, run the following command:

```
# hares -override resource_name IMF
```

```
# hares -modify resource_name IMF -update Mode 0
```

4 Save the VCS configuration.

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

Configuring service groups for SAP NetWeaver

This chapter includes the following topics:

- About configuring service groups for SAP NetWeaver

- Before configuring the service groups for SAP NetWeaver

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

- Configuring service groups for SAP NetWeaver

- Generating an environment file for SAP

- Configuring SAPNW preonline script

About configuring service groups for SAP NetWeaver

Configuring the SAP NetWeaver 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 NetWeaver" on page 84.

Before configuring the service groups for SAP NetWeaver

Before you configure the SAP NetWeaver service group, you must:

- Verify that the Cluster Server components are installed and configured on all nodes in the cluster where you will configure the service group.
  For more information on installing the components, refer to the *InfoScale Availability Installation Guide*.

- Verify that SAP NetWeaver is installed and configured identically on all nodes in the cluster.

- Verify that the Cluster Server agent for SAP NetWeaver is installed on all nodes in the cluster.
  See "Importing the agent types files in a VCS environment" on page 54.

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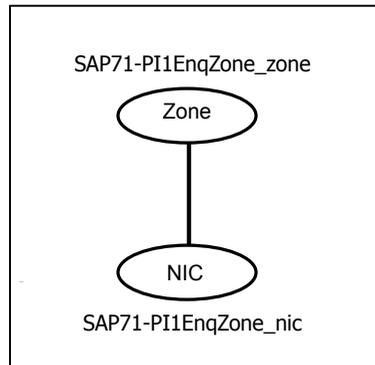
You need not have the IP resource configured for the IP used for zone. When you start the zone, the IP is brought online automatically. When the zone is shut down, the IP is taken offline automatically.

The following figure shows the Properties View for the zone service group.
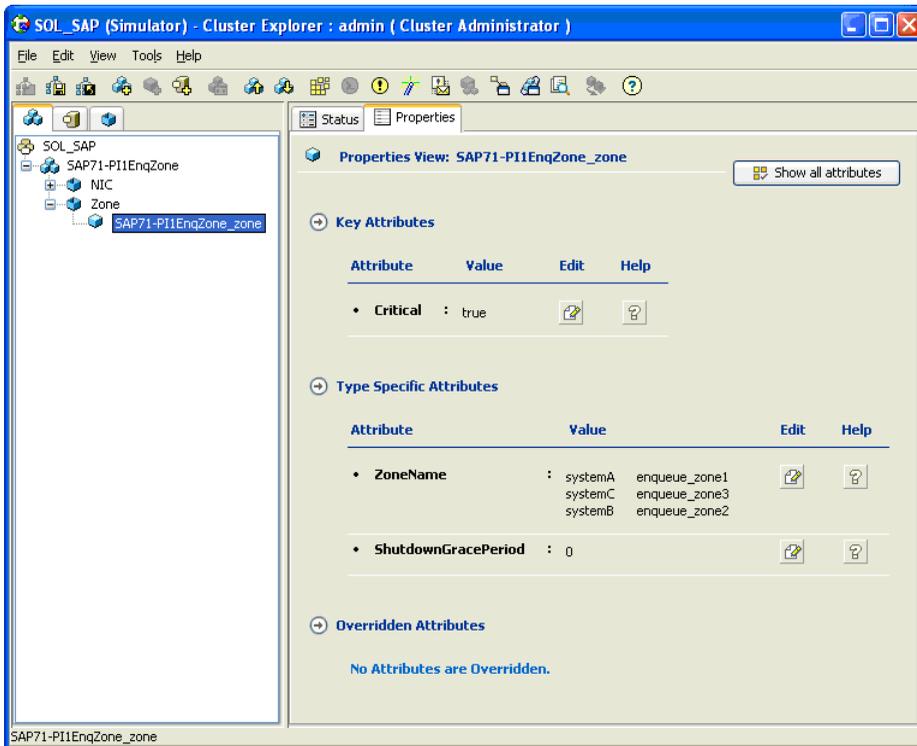
Following is the sample `main.cf` for Zone service group.

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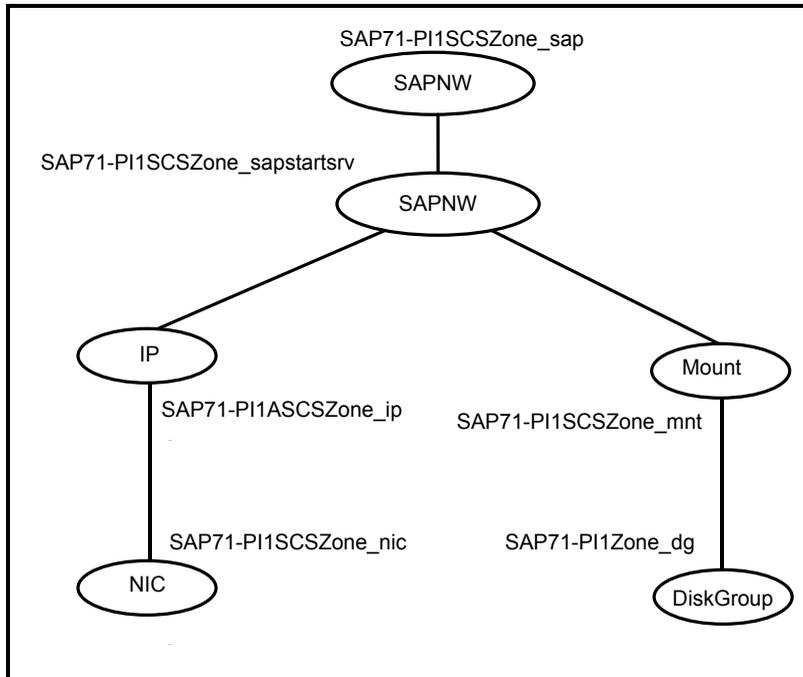
The service group is a failover service group with localized ContainerName attribute for its IP and SAPNW type resources.

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

```
include "types.cf"
include "SAPNW.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
  )

SAPNW 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
  )

SAPNW 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
// {
// // SAPNW SAP71-PI1SCSZone_sap
```

```
// {
//   SAPNW 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.

SAP71-PI1ERSZone_sap

SAPNW

SAP71-PI1ERSZone_sapstartsrv

SAPNW

IP

Mount

SAP71-PI1ERSZone_ip

SAP71-PI1ERSZone_mnt

SAP71-PI1ERSZone_nic

SAP71-PI1ERSZone_dg

NIC

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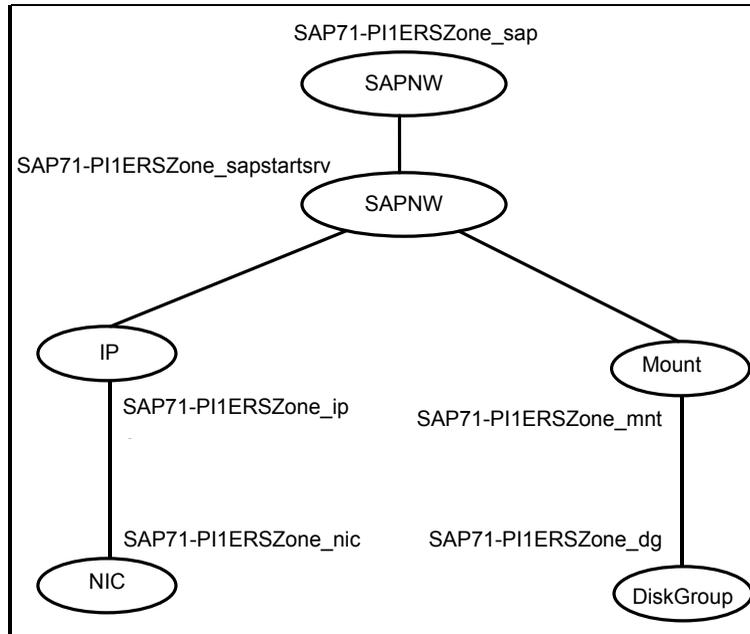
The service group is a failover service group with localized ContainerName attribute for its IP and SAPNW type resources.

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

```
include "types.cf"
include "SAPNW.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
  )


SAPNW 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
  )


SAPNW 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
// {
//   SAPNW SAP71-PI1ERSZone_sap
//     {
//       SAPNW 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 NetWeaver

While setting up a cluster, you must always ensure that the cluster has some spare capacity to handle the SAP NetWeaver 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.

Refer to the Cluster Server installation guide.

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 NetWeaver

1 Create a service group for SAP NetWeaver.

For example,

```
# hagrp -add SAPNW-PI1SCS
```

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

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

For example,

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

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

For example,

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

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

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

4 Create links between the resources. For example,

```
# hares -link SAPNW-PI1SCS_ip SAPNW-PI1SCS_nic
```

5 Create SAPNW resources for SAP. For example,

```
# hares -add SAPNW-PI1SCS_scs SAPNW SAPNW-PI1SCS
```

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

See "SAP NetWeaver agent attributes" on page 55.

6 Create resource dependencies for the SAP NetWeaver resource.

The SAPNW resource depends on IP and Mount resources.

For example,

```
# hares -link SAPNW-PI1SCS_scs SAPNW-PI1SCS_ip
```

7 Verify the final resource dependencies for the SAP NetWeaver service group.

```
# hares -dep
```

For example,

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

Generating an environment file for SAP

Veritas recommends that you use a custom-generated environment file to configure the EnvFile attribute of the agent.

To generate the environment file for SAP applications on non-SystemD platforms

1 Log in as an SAP administrator.

For example:

```
su - pi1adm
```

2 Capture the environment information in a file with the `env` command.

For example:

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

3 Update this file according to the user shell environment of the SAP administrator.

For example, if the generated file contains the environment for the `bash` shell and the SAP administrator uses the C shell, update the file as follows:

- Edit the `sappi1env.env` file to add the "setenv" string at the beginning of each line.

- Replace each instance of the "=" character with a space.

4 Ensure that the file permissions are set appropriately for the SAP administrator.

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

5 Copy the `sappi1env.env` file to a shared directory and set the EnvFile attribute value to this file path.

This file is then used as the environment file for the SAP instance.

To generate the environment file for SAP applications on SystemD-enabled platforms

1 Log in as an SAP administrator.

For example:

```
su - pi1adm
```

2 Capture the environment information in a file with the `env` command.

For example:

```
env > /home/pi1adm/sap_sysd.env
```

3 Ensure that the file permissions are set appropriately for the SAP administrator.

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

4 Copy the `sap_sysd.env` file to a shared directory and set the SystemDEnvFile attribute value to this file path.

This file is then used as the environment file for the SAP instance.

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

The SAPNW 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/SAPNW
	ln -s /opt/VRTSvcs/bin/SAPNW/monitor preonline
5.x	cd /opt/VRTSagents/ha/bin/SAPNW
	ln -s /opt/VRTSagents/ha/bin/SAPNW/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 SAPNW 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 SAPNW Trigger Call here….
#-------------------
# Define variables..
#-------------------
my $sCmd = '/opt/VRTSvcs/bin/SAPNW/preonline';
```

For VCS 5.0, the value of $sCmd must be equal to /opt/VRTSagents/ha/bin/SAPNW/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_SAPNW
5.x	/etc/VRTSagents/ha/conf/SAPNW

- 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*

Troubleshooting the agent for SAP NetWeaver

This chapter includes the following topics:

- Preliminary troubleshooting checks

- Starting the SAP NetWeaver instance outside a cluster

- Reviewing log files

- Configuration checks for Solaris zones support

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

Preliminary troubleshooting checks

If you face problems with the Cluster Server agent for SAP NetWeaver, perform the following checks before further investigation:

- Use the correct software and operating system versions.
  Ensure that no issues arise due to incorrect software and operating system versions. For information on the software versions that the agent for SAP NetWeaver supports, see the Veritas Services and Operations Readiness Tools (SORT) site: https://sort.veritas.com/agents.

- Meet prerequisites.
  Before installing the agent for SAP NetWeaver, double check that you meet the prerequisites. For example, you must install the ACC library on VCS before installing the agent for SAP NetWeaver.

- Configure SAP NetWeaver resources correctly.

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

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 that you 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

■ 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).

■ If you use SAP MMC or the HACheckConfig function of SAPControl to verify the HA configuration of the SAP system, and if you encounter any errors related to the configuration checks, add the following line in ERS profile:

```
enque/enrep/hafunc_implementation = script
```

Then, restart the sapstartsrv process for ERS instance.

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

You can then restart the SAP NetWeaver 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:

```
$ sapcontrol -nr <InstID> -function StartService <SAPSID>
$ sapcontrol -nr <InstID> -function WaitforServiceStarted 300 1
$ sapcontrol -nr <InstID> -function StartWait  300 1
```

Note: The value of the OnlineTimeOut attribute of the SAPNW agent type should be large enough to accommodate the time required to start the SAP instance. If the time required to start the SAP instance exceeds the value specified by the OnlineTimeOut attribute, the SAP instance should be started outside VCS.

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

For example, for an SAP instance:

```
$ 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 log files

If you face problems while using the SAP NetWeaver agent, use the log files described in this section to investigate the problems.

- Cluster log files
  The engine log file is located at `/var/VRTSvcs/log/engine_A.log`. For a long running cluster, the log files are rotated as engine_B.log, engine_C.log, and so on. The most-recent engine logs are present in the engine_A.log file.

- SAP NetWeaver agent log files
  The SAP NetWeaver agent log file is located at `/var/VRTSvcs/log/SAPNW_A.log`.

Note: By default, the VCS log location is `/var/VRTSvcs`. You can customize this log location; the engine logs and the agent logs will be located at your customized log location.

Using trace level logging

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

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

Warning: You may consider temporarily increasing the timeout values for resources 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   Make the cluster writable.

```
# haconf -makerw
```

2   Identify the resource for which you want to enable detailed logging.

3 Localize the ResLogLevel attribute for the identified resource:

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

4 Set the ResLogLevel attribute to TRACE for the identified resource:

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

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

6 Test the identified resource. The function reproduces the problem that you are attempting to diagnose.

7 Note the time when the problem is reproduced.

8 Set the ResLogLevel attribute back to INFO for the identified resource:

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

9 Save the configuration changes.

```
# haconf -dump
```

10 Review the contents of the log file. Use the time noted in Step 4 and Step 6 to diagnose the problem.

For more details, refer to the *Cluster Server Administrator's Guide*. You can also contact Veritas support for more help.

To enable debug logs for all resources of type SAPNW

Enable the debug log.

```
# hatype -modify SAPNW 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 SAPNW LogDbg
# hares -modify SAPNW 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.

See "Configuring SAPNW preonline script" on page 87.

To set the ResLogLevel attribute for preonline trigger

1 Navigate 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/SAPNW/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/SAPNW/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/SAPNW_A.log`.

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 NetWeaver" on page 42.

- Importing the types.cf file for Solaris zone support
  See "Importing the agent types files in a VCS environment" on page 54.

- Configuring the SAP resources for Solaris zone support

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.
  See "Removing the agent" on page 49.

- Boot the zone on the node.

- Install the package on the node.
  See "Installing the agent " on page 46.

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 NetWeaver

- Sample agent type definition

- Sample SAP resource configuration

- Sample service group configuration for ABAP and Java Usage types

- Sample SAP NetWeaver service group configurations for Solaris zone support

- Sample service group dependency

About sample configurations for the agents for SAP NetWeaver

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 NetWeaver. For more information about these resource types, refer to the *Cluster Server Bundled Agents Reference Guide*.

Sample agent type definition

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

An excerpt from this file is as follows:

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

A sample agent type definition file when IMF is enabled is as follows:

```
type SAPNW (
  static int IMF{} = { Mode=2, MonitorFreq=5, RegisterRetryLimit=3 }
  static str IMFRegList[] = { ProcMon, InstProfile }
  static str AgentDirectory = "/opt/VRTSagents/ha/bin/SAPNW"
  static str AgentFile = "/opt/VRTSvcs/bin/Script51Agent"
  static int OfflineTimeout = 600
  static str ArgList[] = { ResLogLevel, State, IState, EnvFile,
    SAPAdmin, InstProfile, InstType, ProcMon, EnqSrvResName,
    MonitorProgram, StopService, UseSystemD, SystemDEnvFile }
  static boolean AEPTimeout = 1
  str ResLogLevel = INFO
  str EnvFile
  str SAPAdmin
  str InstProfile
  str InstType = APPSERV
  str ProcMon
  str EnqSrvResName
  str MonitorProgram
  boolean StopService = 0
  boolean UseSystemD = 0
  str SystemDEnvFile
)
```

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.

An excerpt form the sample `main.cf` file when GCO is configured.

Sample SAP primary application server instance

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

For SAP NetWeaver 7.x with SAP Kernel 7.20 DCK or later with HA-API support

```
SAPNW PRD_PAS_SAPNW_SAPSTARTSRV (
   EnvFile = "/home/prdadm/.login"
   SAPAdmin = prdadm
   InstProfile = "/usr/sap/PRD/SYS/profile/PRD_J01_sapprdapp"
   InstType = SAPSTARTSRV
   ProcMon = "sapstartsrv"
  )
 SAPNW PRD_PAS_SAPNW (
 EnvFile = "/home/prdadm/.login"
 SAPAdmin = prdadm
 InstProfile = "/usr/sap/PRD/SYS/profile/PRD_J01_sapprdapp"
 ProcMon = "jc ig"
  )
```

For SAP NetWeaver versions earlier than 7.x with SAP Kernel 7.20 DCK or later

```
 SAPNW PRD_PAS_SAPNW (
 EnvFile = "/home/prdadm/.login"
 SAPAdmin = prdadm
 InstProfile = "/usr/sap/PRD/SYS/profile/PRD_J01_sapprdapp"
 ProcMon = "jc ig"
 StopService = 1
  )
```

Sample SAP additional application server instance

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

For SAP NetWeaver 7.x with SAP Kernel 7.20 DCK or later with HA-API support

```
SAPNW PRD_AAP_SAPNW_SAPSTARTSRV (
  EnvFile = "/home/prdadm/.login"
  SAPAdmin = prdadm
  InstProfile = "/usr/sap/PRD/SYS/profile/PRD_J03_sapprddi"
  InstType = SAPSTARTSRV
  ProcMon = "sapstartsrv"
  )

 SAPNW PRD_AAP_SAPNW (
  EnvFile = "/home/prdadm/.login"
  SAPAdmin = prdadm
  InstProfile = "/usr/sap/PRD/SYS/profile/PRD_J03_sapprddi"
  ProcMon = "jc  ig"
  )
```

For SAP NetWeaver versions earlier than 7.x with SAP Kernel 7.20 DCK or later

```
 SAPNW PRD_AAP_SAPNW (
  EnvFile = "/home/prdadm/.login"
  SAPAdmin = prdadm
  InstProfile = "/usr/sap/PRD/SYS/profile/PRD_J03_sapprddi"
  ProcMon = "jc  ig"
  StopService = 1
  )
```

Sample SAP Central Services instance

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

For SAP NetWeaver 7.x with SAP Kernel 7.20 DCK or later with HA-API support

```
SAPNW PRD_SCS_SAPNW_SAPSTARTSRV (
  ResLogLevel = TRACE
  EnvFile = "/home/prdadm/.login"
  SAPAdmin = prdadm
  InstProfile = "/usr/sap/PRD/SYS/profile/PRD_SCS00_sapprdscs"
  InstType = SAPSTARTSRV
  ProcMon = "sapstartsrv"
  )
 SAPNW PRD_SCS_SAPNW (
  ResLogLevel = TRACE
  EnvFile = "/home/prdadm/.login"
  SAPAdmin = prdadm
  InstProfile = "/usr/sap/PRD/SYS/profile/PRD_SCS00_sapprdscs"
```

```
InstType = ENQUEUE
ProcMon = "en ms "
)
```

For SAP NetWeaver versions earlier than 7.x with SAP Kernel 7.20 DCK or later

```
SAPNW PRD_SCS_SAPNW (
ResLogLevel = TRACE
EnvFile = "/home/prdadm/.login"
SAPAdmin = prdadm
InstProfile = "/usr/sap/PRD/SYS/profile/PRD_SCS00_sapprdscs"
InstType = ENQUEUE
ProcMon = "en ms "
StopService = 1
)
```

Sample SAP Enqueue Replication server instance

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

For SAP NetWeaver 7.x with SAP Kernel 7.20 DCK or later with HA-API support

```
SAPNW PRD_ERS_SAPNW_SAPSTARTSRV (
EnvFile = "/home/prdadm/.login"
SAPAdmin = prdadm
InstProfile = "/usr/sap/PRD/SYS/profile/PRD_ERS10_sapprders"
InstType = SAPSTARTSRV
ProcMon = "sapstartsrv"
EnqSrvResName = PRD_SCS_SAPNW
UseSystemD = 1
SystemDEnvFile = "/home/tstadm/login.env1"
)

SAPNW PRD_ERS_SAPNW (
EnvFile = "/home/prdadm/.login"
SAPAdmin = prdadm
InstProfile = "/usr/sap/PRD/SYS/profile/PRD_ERS10_sapprders"
InstType = ENQREP
ProcMon = er
EnqSrvResName = PRD_SCS_SAPNW
UseSystemD = 1
SystemDEnvFile = "/home/tstadm/login.env1"
)
```

For SAP NetWeaver versions earlier than 7.x with SAP Kernel 7.20 DCK or later

```
SAPNW PRD_ERS_SAPNW (
  EnvFile = "/home/prdadm/.login"
  SAPAdmin = prdadm
  InstProfile = "/usr/sap/PRD/SYS/profile/PRD_ERS10_sapprders"
  InstType = ENQREP
  ProcMon = er
  EnqSrvResName = PRD_SCS_SAPNW
  StopService = 1
  UseSystemD = 1
  SystemDEnvFile = "/home/tstadm/login.env1"
)
```

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).

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

Sample service group configurations for SAP NetWeaver with HA-API support

To configure high availability for SAP NetWeaver 7.x with SAP Kernel 7.20 DCK or later with HA-API support, you must set the StopService attribute to 0 and you must configure an application resource.

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

Figure A-1   Service group configuration for a Primary Application Server
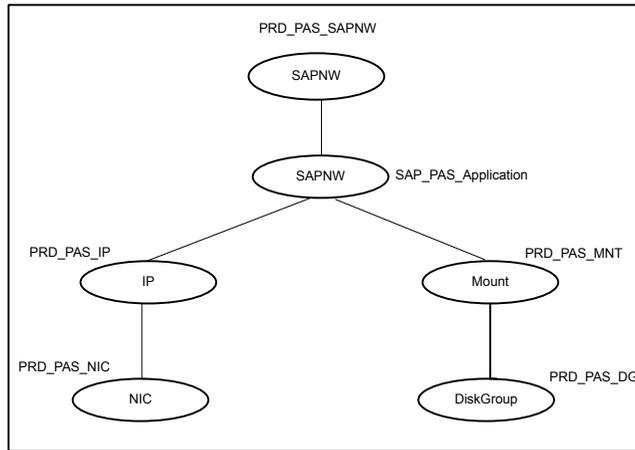

Figure A-2 shows a sample service group configuration for an Additional Application Server.

**Figure A-2**   Service group configuration for an Additional Application Server
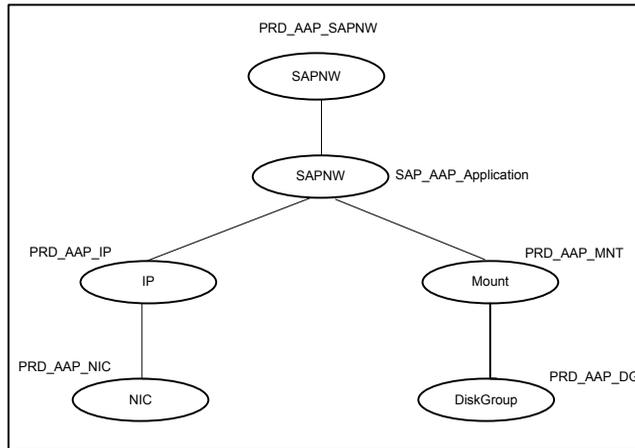
Figure A-3 shows a sample service group configuration for an SAP Central Services.

Figure A-3 Service group configuration for SAP Central Services
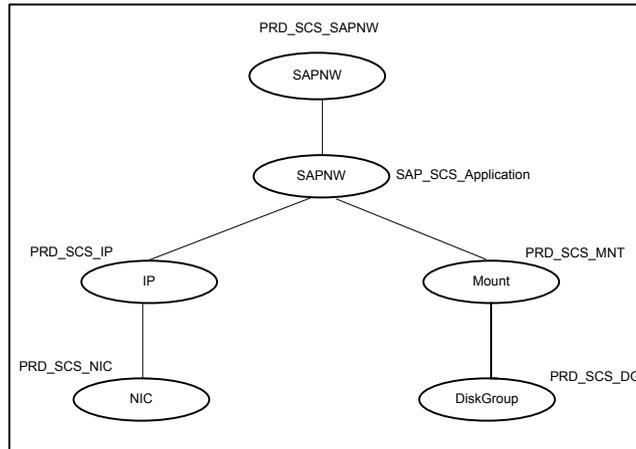
Figure A-4 shows a sample service group configuration for Enqueue Replication Server instance.

Figure A-4 Service group configuration for Enqueue Replication Server instance
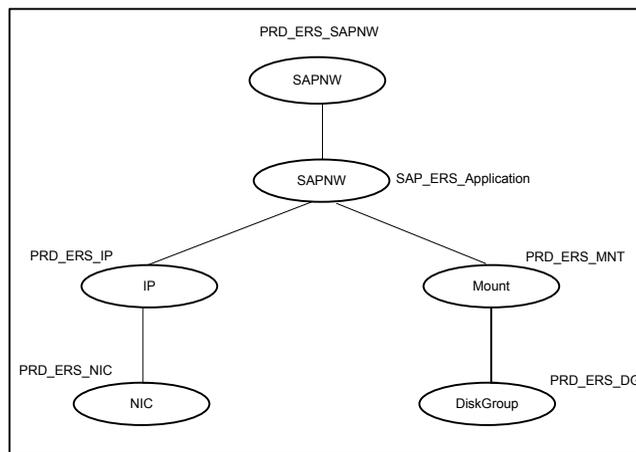
Sample service group configurations for versions earlier than SAP NetWeaver 7.x with SAP Kernel 7.20 DCK or later

Figure A-5 shows a sample service group configuration for a Primary Application Server.

Figure A-5   Service group configuration for a Primary Application Server
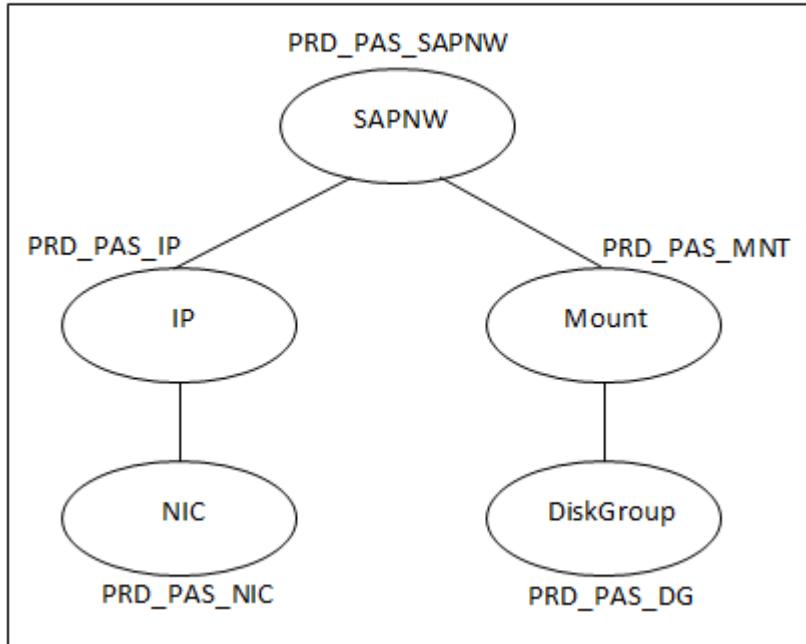
Figure A-6 shows a sample service group configuration for an Additional Application Server.

Figure A-6   Service group configuration for an Additional Application Server
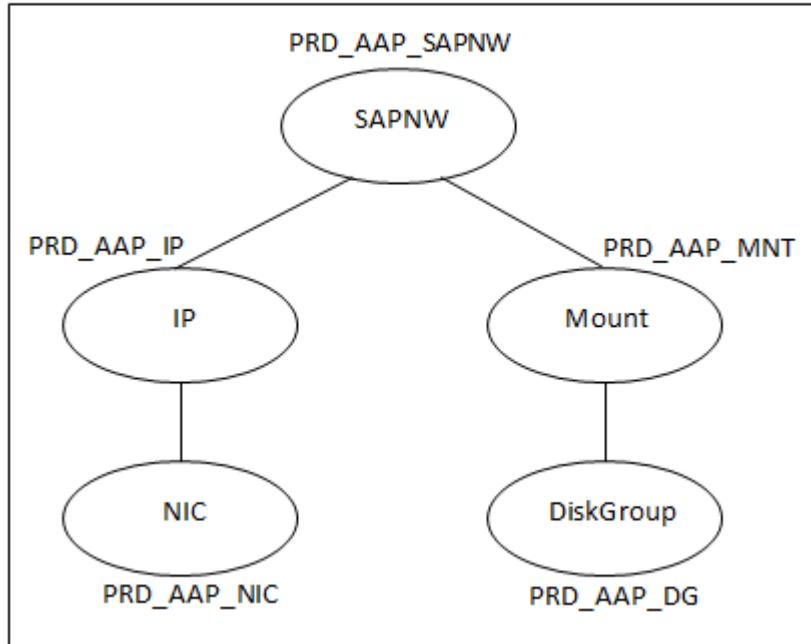

Figure A-7 shows a sample service group configuration for an SAP Central Services.

**Figure A-7**   Service group configuration for SAP Central Services
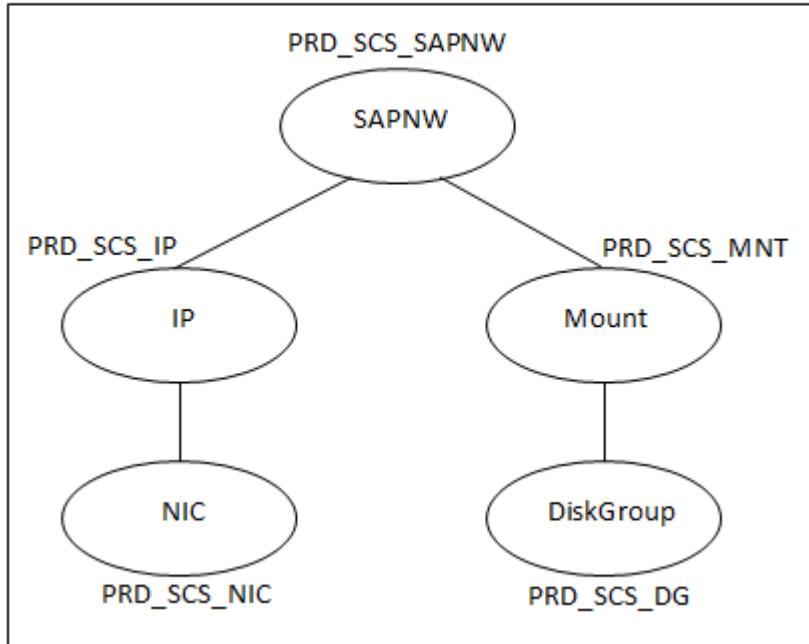
Figure A-8 shows a sample service group configuration for Enqueue Replication Server instance.

Figure A-8   Service group configuration for Enqueue Replication Server
instance
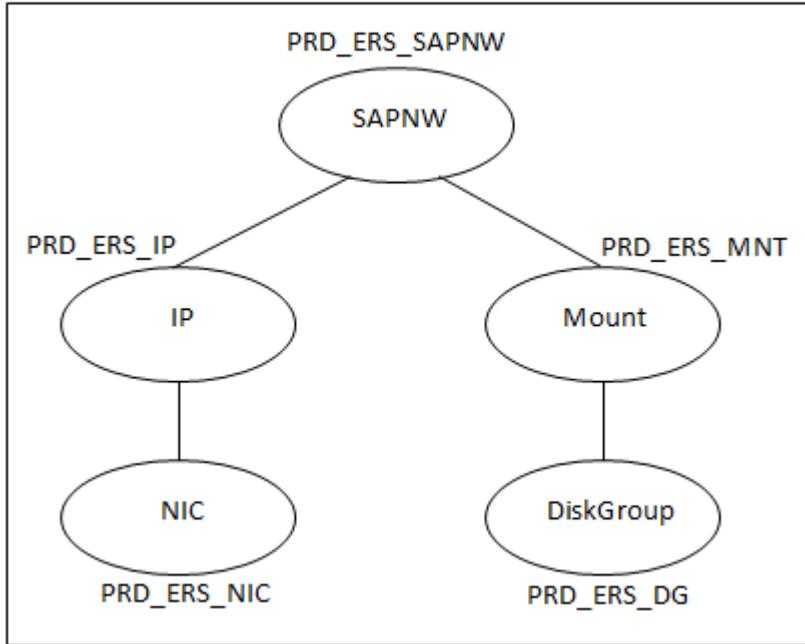
Sample SAP NetWeaver service group configurations for Solaris zone support

This section includes sample service groups with Solaris zone support.

Figure A-9 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-9   Service group with loop back file systems for application server instance running in a non-global zone
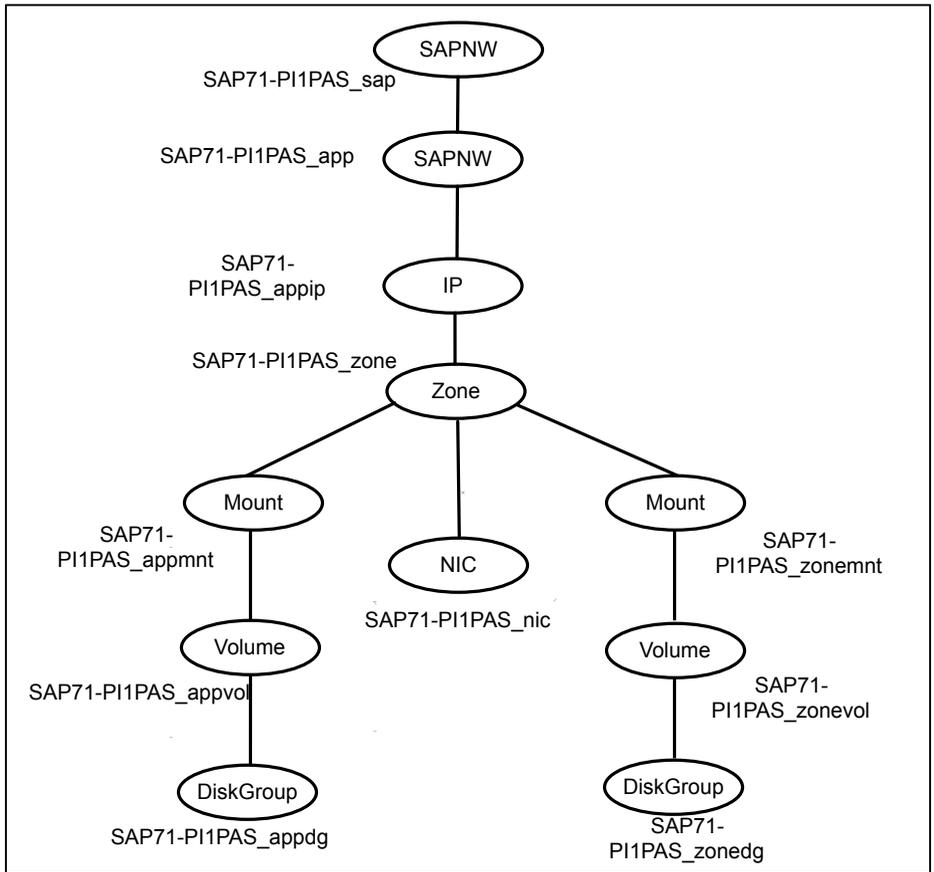
Sample service group dependency

Figure A-10 shows the sample service group dependency for SAP NetWeaver.

Figure A-10   Sample service group dependency