Veritas™ High Availability Agent for SAP WebAS Installation and Configuration Guide

HP-UX, Linux, Solaris

5.0

March 2009

symantec™

Veritas High Availability Agent for SAP WebAS Installation and Configuration Guide

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

Agent version: 5.0.1.0

Document version: 5.0.1

Legal Notice

Symantec Corporation
20330 Stevens Creek Blvd.
Cupertino, CA 95014

http://www.symantec.com

Technical Support

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

Symantec's maintenance offerings include the following:

- A range of support options that give you the flexibility to select the right amount of service for any size organization
- Telephone and Web-based support that provides rapid response and up-to-the-minute information
- Upgrade assurance that delivers automatic software upgrade protection
- Global support that is available 24 hours a day, 7 days a week
- Advanced features, including Account Management Services

For information about Symantec's Maintenance Programs, you can visit our Web site at the following URL:

www.symantec.com/techsupp/

Contacting Technical Support

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

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

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

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

- Product release level
- Hardware information
- Available memory, disk space, and NIC information
- Operating system

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

Licensing and registration

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

www.symantec.com/techsupp/

Customer service

Customer service information is available at the following URL:

www.symantec.com/techsupp/

Customer Service is available to assist with the following types of issues:

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

Documentation feedback

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

Maintenance agreement resources

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

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

Additional enterprise services

Symantec offers a comprehensive set of services that allow you to maximize your investment in Symantec products and to develop your knowledge, expertise, and global insight, which enable you to manage your business risks proactively.

Enterprise services that are available include the following:

Symantec Early Warning Solutions	These solutions provide early warning of cyber attacks, comprehensive threat analysis, and countermeasures to prevent attacks before they occur.
Managed Security Services	These services remove the burden of managing and monitoring security devices and events, ensuring rapid response to real threats.
Consulting Services	Symantec Consulting Services provide on-site technical expertise from Symantec and its trusted partners. Symantec Consulting Services offer a variety of prepackaged and customizable options that include assessment, design, implementation, monitoring, and management capabilities. Each is focused on establishing and maintaining the integrity and availability of your IT resources.
Educational Services	Educational Services provide a full array of technical training, security education, security certification, and awareness communication programs.

To access more information about Enterprise services, please visit our Web site at the following URL:

www.symantec.com

Select your country or language from the site index.

Contents

Chapter 1

Introducing the Veritas High Availability Agent for SAP WebAS

This chapter includes the following topics:

About the Veritas agent for SAP WebAS

The Veritas High Availability agents monitor specific resources within an enterprise application. They determine the status of resources and start or stop them according to external events.

The Veritas agent for SAP WebAS provides high availability for SAP NetWeaver 7.1 in a cluster. The agent for SAP WebAS is designed to support a wide range of SAP NetWeaver environments which include SAP NetWeaver CE 7.1, SAP NetWeaver Mobile 7.1, and SAP NetWeaver PI 7.1.

See the following Technical Support TechNote for the latest updates or software issues for this agent:

http://seer.entsupport.symantec.com/docs/282004.htm

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

The agent supports the following SAP instance types:

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

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

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

What's new in this agent

The enhancements in this release of Veritas High Availability agent for SAPWebAS are as follows:

- Added support for RHEL, SUSE and HP-UX.

Supported software

The Veritas agent for SAP WebAS supports the following software versions:

Veritas Cluster Server	VCS 4.0, 4.1, 5.0 Apply appropriate Maintenance Pack patches to VCS to support mount agent for NFS mounts inside Solaris non-global zones.
ACC Library	5.1.4.0 and later Review the ACC Library version for i18n support. See "Prerequisites for enabling i18n support" on page 22.
Operating Systems	Solaris 9, 10 SPARC Red Hat Enterprise Linux 4.0, 5.0 SUSE Linux 9.0, 10.0 HP-UX 11iv2, 11iv3
SAP WebAS	7.1

SAP Applications	NetWeaver PI 7.1, Mobile 7.1, and CE 7.1 **Note:** All Enhancement Packages (EhP) for PI 7.1, Mobile 7.1 and CE 7.1 are supported.

About SAP Web Application Server

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

The following three usage types are possible with SAP WebAS:

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

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

SAP system components

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

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

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

SAP architecture

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

Table 1-1 SAP architecture

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

Table 1-1 SAP architecture *(continued)*

Architecture	Component	Service	Functions
	Enqueue Replication Instance	Java Enqueue Replication Service	Enables the lock table to be replicated on a second server, the replication server.
SAP WebAS Add-In	Application Server	ABAP Dispatcher	■ Controls program that manages the resources of R/3 applications. ■ Balances the assignments of the transaction load to the work processes. ■ Manages buffer in main memory. ■ Connects to the presentation level. ■ Organizes the communication processes.
		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.

Table 1-1 SAP architecture *(continued)*

Architecture	Component	Service	Functions
	Enqueue Replication Instance ABAP	ABAP Enqueue Replication Service	Enables the lock table to be replicated on a second server, the replication server.
	Enqueue Replication Instance Java	Java Enqueue Replication Service	Enables the lock table to be replicated on a second server, the replication server.

Single Point of Failures (SPOF)

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

- Database Instance
- Central Services Instance
- Enqueue Replication Server
- Network File System

Table 1-2 lists the possibilities to eliminate the single point of failures.

Table 1-2 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

SAP WebAS agent functions

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

Online

The online function performs the following tasks:

- Performs a preliminary check to ensure that the SAP instance is not online on the specified node in the cluster.
- Removes any SAP processes that remain because of an unclean shutdown as follows:
  - If the SAP instance is APPSERV or ENQREP, the `cleanipc` utility gets executed. Otherwise, the agent kills all relevant SAP processes.
  - If the `kill.sap` and `shutdown.sap` file exists in the /usr/sap/SAPSID/InstName/work directory, the function removes the file from the directory.
  - Removes the SE and CO locks files from the /usr/sap/SAPSID/InstName/data directory.
- Initiates the standard SAP error log process.
- Starts the sapstartsrv process for Web-based SAP Management console.
- Starts the SAP instance using the `sapstart` command.
- Ensures that the instance is fully initialized.

Offline

The offline function performs the following tasks:

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

Monitor

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

- Depending upon the search criteria that the ProcMon attribute specifies, the monitor function scans the process table to verify the SAP instance processes are running. For more information about setting the ProcMon attribute:
See "Monitoring an SAP instance" on page 36.
- If the SecondLevelMonitor attribute is greater than 0, the monitor function performs a thorough health check of the SAP instance as follows:
  - For APPSERV instances, the function uses the following utilities to perform this check:

Server type	SAP utility used
SAP Web Application Server as ABAP	sapinfo
SAP Web Application Server as Java	jsmon
SAP Web Application Server as Add-In	sapinfo and jsmon

  - For Enqueue Server and Enqueue Replication Server instances, the function uses the `ensmon` utility.
- The monitor function executes a custom monitor utility.
See "Executing a customized monitoring program" on page 39.

Clean

The clean function performs the following tasks:

- Sends a SIGINT signal to the `sapstart` process, if the process exists. Otherwise, the function sends a SIGINT signal to all running processes that are relevant to the specified SAP instance.
- Ensures that no relevant SAP processes are running. If any processes remain, the operation kills all the remaining processes using a SIGKILL signal.
- If the `kill.sap` and/or `shutdown.sap` file exists in the /usr/sap/SAPSID/InstName/work directory, the operation removes the file from the directory

- Removes the SE and CO lock files from the /usr/sap/SAPSID/InstName/data directory.
- If the SAP Instance is APPSERV or ENQREP, the operation executes the `cleanipc` utility.
- Augments the SAP log.

Chapter 2

Installing, upgrading, and removing the agent for SAP WebAS

This chapter includes the following topics:

Before you install the Veritas agent for SAP WebAS

Ensure that you meet the following prerequisites to install the agent for SAP WebAS.

For VCS, do the following:

- Install and configure Veritas Cluster Server.
  For more information on installing and configuring Veritas Cluster Server refer to, `Veritas Cluster Server Installation Guide`
- Remove any previous version of this agent.
  To remove the agent,
  See "Removing the agent in a VCS environment" on page 24.

- Install the latest version of ACC Library.
  To install or update the ACC Library package, locate the library and related documentation on the agentpack disc.
  See "Installing the ACC library" on page 23.

Prerequisites for enabling i18n support

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

- Install ACCLib version 5.1.4.0 or later.
  You can find the latest version of ACCLib on the agent pack disc at the following location:

Platform	Location
HP-UX	cd1/hpux/generic/application/acc_library/vcs/version_library/pkgs/
Linux	cd1/linux/generic/application/acc_library/vcs/version_library/rpms/
Solaris	cd1/solaris/*arch_dist*/application/acc_library/vcs/version_library/pkgs/ where *arch_dist* takes the value 'sparc'

- Copy the latest ag_i18n_inc.pm module from the following location on the agent pack disc.

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

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

  where *arch_dist* takes the following values:
  'sparc' for Solaris SPARC
  'generic' for Linux and HP-UX

Prerequisites for installing the agent to support Solaris zones

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

- Install SAP to support Solaris zones. Refer to the SAP note 870652.

- Install and configure the VCS 5.0 environment to support Solaris zones. Refer to the VCS user documentation for details.
- Remove any previous version of this agent.

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

About ACC Library

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

Installing the ACC library

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

To install the ACC library

1 Log in as superuser.

2 Navigate to the pkgs directory (the pkgs directory on the CD).

HP-UX	*cd_mount*/hpux/generic/application/acc_library/vcs/*version*_library/pkgs
Linux	*cd_mount*/linux/generic/application/acc_library/vcs/*version*_library/rpms
Solaris	*cd_mount*/solaris/*dist_arch*/application/acc_library/vcs/*version*_library/pkgs where *dist_arch* is sparc

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

HP-UX	`# swinstall -s 'pwd' VRTSacclib`
Linux	`# rpm -i \` `VRTSacclib-`*`VersionNumber`*`-GA_GENERIC.noarch.rpm`
Solaris	`# pkgadd -d . VRTSacclib`

Installing the agent in a VCS environment

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

To install the agent

1. Log in as superuser.
2. Navigate to the directory containing the package for the platform running in your environment.

   HP-UX `cd_mount/hpux/generic/application/sapwebas_agent/vcs_version/version_agent/pkgs`

   Linux `cd_mount/linux/generic/application/sapwebas_agent/vcs_version/version_agent/rpms`

   Solaris `cd_mount/solaris/dist_arch/application/sapwebas_agent/vcs_version/version_agent/pkgs`

   Where *dist* is the Solaris distribution and *arch* is the Solaris processor architecture.

3. Install the package.

   HP-UX
   ```
   # swinstall -s 'pwd' VRTSsapwebas71
   ```

   Linux
   ```
   # rpm -ihv \
   VRTSsapwebas71-AgentVersion-GA_GENERIC.noarch.rpm
   ```

   Solaris
   ```
   # pkgadd -d . VRTSsapwebas71
   ```

Removing the agent in a VCS environment

You must uninstall the agent for SAP WebAS from a cluster while the cluster is active.

To uninstall the agent in a VCS environment

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

   ```
   # haconf -makerw
   ```

3. Remove all SAP Web AS resources from the cluster. Use the following command to verify that all resources have been removed:

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

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

   ```
   # hatype -delete SAPWebAS71
   ```

   Removing the agent's type file from the cluster removes the include statement for the agent from the main.cf file, but the agent's type file is not removed from the cluster configuration directory. You can remove the agent's type file later from the cluster configuration directory.
5. Save these changes. Then set the cluster configuration mode to read-only by typing the following command from any node in the cluster:

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

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

   Execute the following command to uninstall the agent:

HP-UX	`# swremove VRTSsapwebas71`
Linux	`# rpm -e VRTSsapwebas71`
Solaris	`# pkgrm VRTSsapwebas71`

Removing the ACC library

Perform the following steps to remove the ACC library.

To remove the ACC library

1. Ensure that all agents that use ACC library are removed.
2. Run the following command to remove the ACC library package.

HP-UX	`# swremove VRTSacclib`
Linux	`# rpm -e VRTSacclib`
Solaris	`# pkgrm VRTSacclib`

Upgrading the agent in a VCS environment

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

1. Persistently freeze the service groups that host the application.

   `# hagrp -freeze` *`group`* `-persistent`

2. Stop the cluster services forcibly.

   `# hastop -all -force`

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

   `# ps -ef |grep SAPWebAS71`

4. Uninstall the agent package from all the nodes.

   See "Removing the agent in a VCS environment" on page 24.

5. Install the new agent on all the nodes.

   See "Installing the agent in a VCS environment" on page 24.

6. Copy the new SAPWebAS71Types.cf file from the agent's sample conf directory,

VCS 4.x	/etc/VRTSagents/ha/bin/SAPWebAS71
VCS 5.0	/etc/VRTSvcs/conf/sample_SAPWebAS71

   to the VCS conf directory /etc/VRTSvcs/conf/config.

Note: If you are using Solaris Zones, copy the SAPWebAS71Types_zones.cf file from the agent's sample conf directory.

---

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

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

8 Start the cluster services.

```
# hastart
```

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

```
# haagent -start SAPWebAS71 -sys System
```

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

```
# hagrp -unfreeze group -persistent
```

Chapter 3

Preparing to configure the agent for SAP WebAS

This chapter includes the following topics:

About configuring the Veritas agent for SAP WebAS

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

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

See "About sample configurations for the agent for SAP WebAS" on page 83.

Importing the agent types files in a VCS environment

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

To import the agent types file using the Veritas Cluster Server graphical user interface

1. Start the Veritas 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 4.x	/etc/VRTSvcs/conf/sample_SAPWebAS71/SAPWebAS71Types.cf
VCS 5.0	/etc/VRTSagents/ha/conf/SAPWebAS71/SAPWebAS71Types.cf
For Solaris zone support	/etc/VRTSagents/ha/conf/SAPWebAS71/SAPWebAS71Types_zones.cf

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

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

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

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

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

   ```
   # haconf -makerw
   ```

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

VCS 4.x	# cd /etc/VRTSvcs/conf/sample_SAPWebAS71
VCS 5.0	# cd /etc/VRTSagents/ha/conf/SAPWebAS71

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

For example,

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

4 Execute the following file.

./SAPWebAS71Types.cmd

Note: For Solaris zones, execute the # ./SAPWebAS71Types_zones.cmd file.

5 Save the configuration.

`haconf -dump -makero`

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

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

SAP WebAS agent attributes

Table 3-1 shows the required attributes for configuring a SAP WebAS instance.

Table 3-1 Required attributes

Required attributes	Description
EnqSrvResName	The name of the VCS resource for SAP Central Services (A)SCS Instance. This attribute is used by Enqueue and Enqueue Replication Server. Using this attribute the Enqueue server queries the Enqueue Replication Server resource state while determining the fail over target and vice a versa. Type and dimension: string-scalar Default: "" Example: SAP71-PI1SCS_sap

Table 3-1 Required attributes *(continued)*

Required attributes	Description
EnvFile	The absolute path to the file that must be sourced with the UNIX shell. You must source this file to set the environment before executing SAP scripts for online, offline, monitor, and clean operations. Supported shell environments are ksh, sh, and csh. **Note:** Ensure that the syntax of this file is in accordance with the user shell that the SAPAdmin attribute specifies. Review the information on how to generate environments file for SAP. See "Generating environments file for SAP" on page 34. Symantec recommends that you store this file on shared disk so that the file is always available to an online system. Type and dimension: string-scalar Default: "" Example: /usr/sap/PI1/DVEBMGS00/sappi1.env
InstType	An identifier that classifies and describes the SAP server instance type. Valid values are: ■ APPSERV: SAP Application Server ■ ENQUEUE: SAP Central Services ■ EBQREP: Enqueue Replication Server **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 36. Type and dimension: string-scalar Default: "" Example: dw se jstart

Table 3-1 Required attributes *(continued)*

Required attributes	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
SAPAdmin	SAP System administrator for SAPSID. This user name is usually a concatenation of the SAPSID attribute and the adm string 'sidadm'. This user name is stored in one or more system naming services, for example, NIS, NIS+, and LDAP servers. The agent operations use this user name to execute their respective core subroutines. Type and dimension: string-scalar Default: "" Example: pi1adm
InstProfile	The full path to the SAP Instance profile. The *SAPSID* is found in /usr/sap/*SAPSID*/SYS/profile directory. The value of the instance is SAPSID_*InstName_hostname*. The hostname must resolve into a valid IP address that is used to cluster the SAP instance. Type and dimension: string-scalar Default: "" Example: /usr/sap/PI1/SYS/profile/PI1_DVEBMGS00_sappi1pas

Table 3-2 lists the optional attributes.

Table 3-2 Optional attributes

Optional attribute	Description
MonitorProgram	Absolute path name of an external, user-supplied monitor executable. Review the information about setting this attribute: ■ See "Executing a customized monitoring program" on page 39. ■ See "Setting the SecondLevelMonitor attribute" on page 37. Type and dimension: string-scalar Default: "" Example 1: /usr/sap/PI1/DVEBMGS00/work/myMonitor.sh Example 2: /usr/sap/PI1/DVEBMGS00/work/myMonitor.sh arg1 arg2
SecondLevelMonitor	Used to enable second-level monitoring. Second-level monitoring is a deeper, more thorough state check of the SAP instance. The numeric value specifies how often the monitoring routines must run. 0 means never run the second-level monitoring routines, 1 means run routines every monitor interval, 2 means run routines every second monitor interval, and so on. **Note:** Exercise caution while setting SecondLevelMonitor to large numbers. For example, if the MonitorInterval is set to 60 seconds and the SecondLevelMonitor is set to 100, then sapinfo is executed every 100 minutes, which may not be as often as intended. For maximum flexibility, no upper limit is defined for SecondLevelMonitor. Type and dimension: integer-scalar Example: 1 Default: 0
ContainerName	Non-global zone support for Solaris. Defines the name of the non-global zone. For more details refer to *Veritas Cluster Server User's Guide* Type and dimension: string-scalar Default: " " Example: sap-zone1

Generating environments file for SAP

Symantec recommends using a custom generated environments file to configure the EnvFile attribute of the SAPWebAS71 agent. The steps to generate the environments file for SAP applications are given as follows.

To generate the environments file for SAP applications

1 Login as *SAPAdmin* user.

```
su – pi1adm
```

2 Capture the environment with the following command.

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

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

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

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

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

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

Uniquely identifying SAP server instances

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

For multiple instances running concurrently on a single node, the agent must be able to uniquely identify each SAP WebAS 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 00-97. 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

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 3-3 lists valid values of the ProcMon attribute

Table 3-3 Values of ProcMon attribute

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

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

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

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

Setting the SecondLevelMonitor attribute

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

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

- For ABAP application Server: sapinfo
  sapinfo is not a standard binary shipped by SAP with installation media. Customers need to download the latest rfcsdk kit from the following site: http: //service.sap.com/swdc -> Support Packages and Patches -> Entry by Application Group -> Additional Components.
  For more information on selecting the right RFCSDK for your SAP application, refer to SAP notes 1005832, 825494 and 413708.
  Copy the sapinfo binary and the needed libraries, if any, to SAP Instance specific directory /usr/sap/<SAPSID>/<InstName>/exe and ensure that the binary and the libraries are specified in the sapcpe binary list for the SAP Instance, by copying sapinfo binary and the needed libraries, if any, to /sapmnt/<SAPSID>/exe.
- For Java application server: jsmon
- For Add-In (ABAP + Java) application server: sapinfo, jsmon
- For Enqueue and Enqueue Replication Server: ensmon

Configuring the SAP WebAS agent for message server restart

In case the message server process fails, the Veritas High Availability agent for SAP WebAS 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 Veritas High Availability agent for SAP WebAS.

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

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

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

```
Restart_Program_00 = local $(_MS)

pf=$(DIR_PROFILE)/PI1_ASCS00_sappi1scs
```

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

The default signal mask consists of the following signals:

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

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

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

Note: Symantec recommends to carefully study the SAP note before you modify the profile files for (A)SCS instance.

Executing a customized monitoring program

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

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

- The specified utility is a valid executable file.
- The first level process check indicates that the SAP Web AS instance is online.
- The SecondLevelMonitor attribute is either set to 0 or 1, and the second level check indicates that the SAP Web AS instance is online.
- The SecondLevelMonitor attribute is set to greater than 1, but the second level check is deferred for this monitoring cycle.

The monitor function interprets the utility exit code as follows:

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

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

Preventing early faulting of Java and Add-in instances

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

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

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

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

The Veritas High Availability agent for SAP WebAS 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 Veritas Cluster Server to run under Solaris non-global zones.

Before configuring VCS in the non-global zones, create a service group for the zone resource similar to the resources and dependencies shown in the following sections.

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.

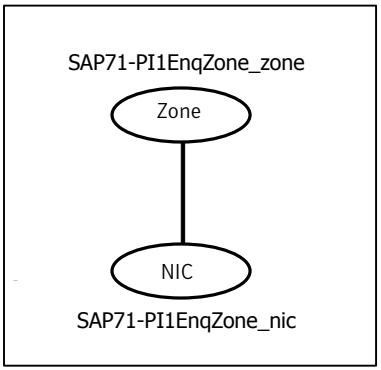
Creating service groups for Enqueue and Enqueue Replication Server under Solaris non-global zones

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

1 Configure a parallel service group for zone resource.

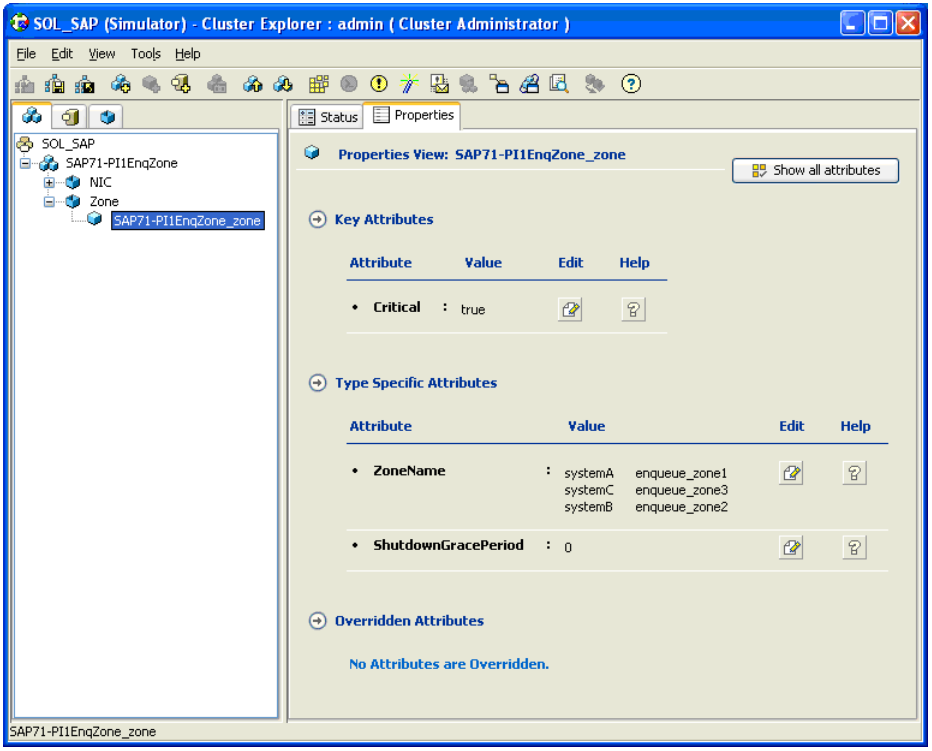
The following figure shows the zone service group configuration for Enqueue and Enqueue Replication Server. This service group is a parallel service group with localized ZoneName attribute for each cluster system.

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 *Veritas Cluster Server User'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 XXXXX vcssx074`

- 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.

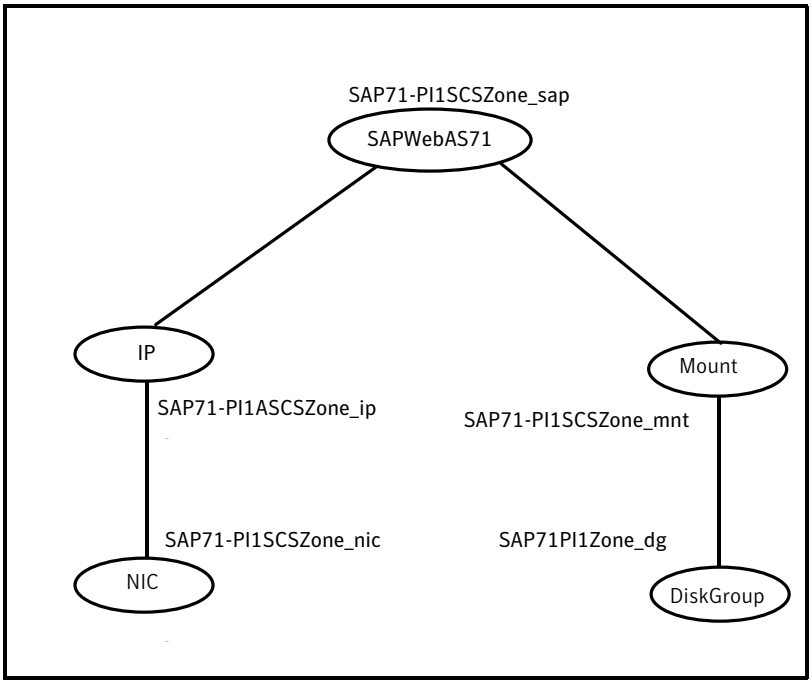
Make sure that the zone configuration files are consistent on all nodes at all times. The file is located at /etc/zones/zone_name.xml.

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.

2 Create the service group for Enqueue Server.

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

The following figure shows the resource dependencies for Enqueue Server.



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

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

```
include "types.cf"
include "SAPMaxDBTypes.cf"
include "SAPWebAS71Types_zones.cf"

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

system systemA (
```

```
  )

system systemB (
  )

system systemC (
  )

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

DiskGroup SAP71-PI1SCSZone_dg (
  DiskGroup = sappi1scs_dg
  )

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

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

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

SAPWebAS71 SAP71-PI1SCSZone_sap (
  EnvFile = "/home/pi1adm/sappi1.env"
  InstProfile = "/usr/sap/PI1/SYS/profile/PI1_SCS20_sappi1scs"
  InstType = ENQUEUE
  ProcMon = "ms en"
  SAPAdmin = pi1adm
```

```
  ContainerName @systemA = enqueue_zone1
  ContainerName @systemB = enqueue_zone2
  ContainerName @systemC = enqueue_zone3
  )

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

// resource dependency tree
//
// group SAP71-PI1SCSZone
// {
// SAPWebAS71 SAP71-PI1SCSZone_sap
//              {
//             Mount SAP71-PI1SCSZone_mnt
//                      {
//                      DiskGroup SAP71-PI1SCSZone_dg
//                       }
//             IP SAP71-PI1SCSZone_ip
//                      {
//                      NIC SAP71-PI1SCSZone_nic
//                      }
//             }
// }
```

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,

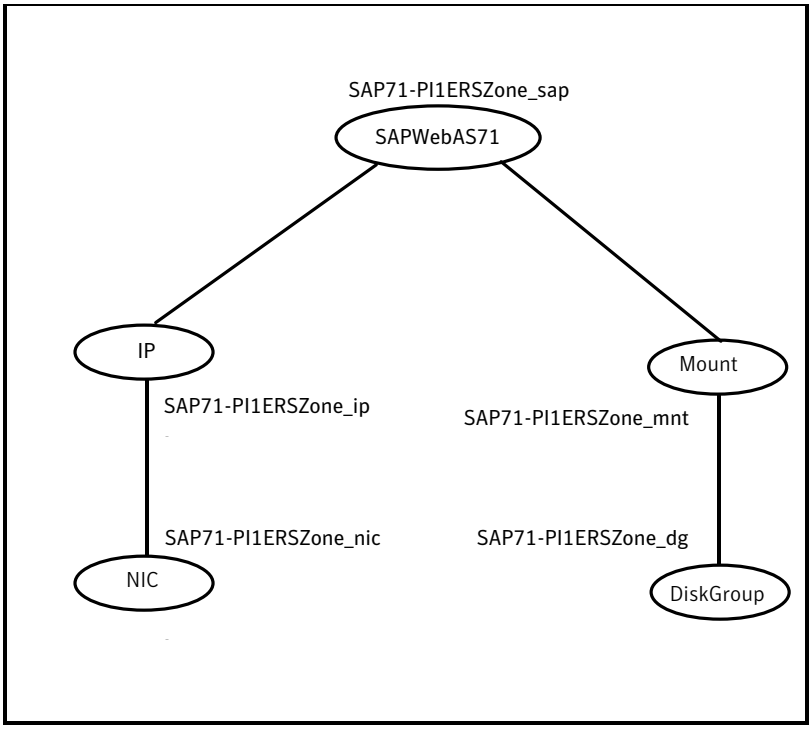
```
# hazoneverify SAP71-PI1SCSZone
```

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

4 Create the service group for Enqueue Replication Server.

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

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



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

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

```
include "types.cf"
include "SAPMaxDBTypes.cf"
include "SAPWebAS71Types_zones.cf"

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

```
 )

system systemA (
  )

system systemB (
  )

system systemC (
  )

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

DiskGroup SAP71-PI1ERSZone_dg (
  DiskGroup = sappi1ers_dg
  )

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

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

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

SAPWebAS71 SAP71-PI1ERSZone_sap (
  EnvFile = "/home/pi1adm/sappi1.env"
  InstProfile = "/usr/sap/PI1/SYS/profile/PI1_ERS21_sappi1ers"
```

```
  InstType = ENQREP
  ProcMon = "er"
  SAPAdmin = pi1adm
  ContainerName @systemA = enqueue_zone1
  ContainerName @systemB = enqueue_zone2
  ContainerName @systemC = enqueue_zone3
  )

requires group SAP71-PI1EnqZone online local firm
SAP71-PI1ERSZone_mnt requires SAP71-PI1ERSZone_dg
SAP71-PI1ERSZone_ip requires SAP71-PI1ERSZone_nic
SAP71-PI1ERSZone_sap requires SAP71-PI1ERSZone_mnt
SAP71-PI1ERSZone_sap requires SAP71-PI1ERSZone_ip

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

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.

Mounting NFS file system inside Solaris non-global zone

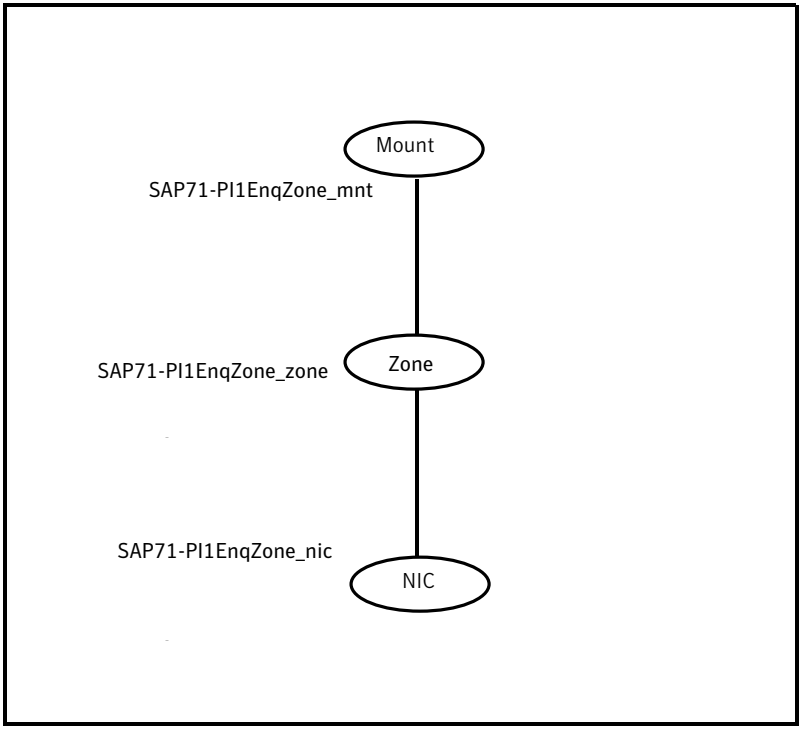
For SAP to function inside Solaris non-global zones the SAP central file system must be available inside the zone. To achieve this, share the SAP central file system on all the client machines using Network File Systems (NFS). One system exports and shares the central file system and others access the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

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

To overcome this issue, you must mount the central file system directly inside the non-global zone using NFS. Alternatively, you can use Veritas 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 3-1 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
//                 }
//         }
// }
```

Chapter 4

Configuring the service groups for SAP Web AS

This chapter includes the following topics:

Configuring SAP WebAS for high availability

The guidelines for configuring SAP WebAS 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 envionment. Since 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 service groups for SAP WebAS

A cluster must have spare capacity to handle SAP instance failover scenarios. For example, in case of a backend database failure, the cluster must be able to run another database instance, in conjunction with other running applications. Review the information about data protection.

See *Veritas Cluster Server Installation and Configuration Guide*.

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

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

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

1. Create a service group for SAP WebAS.

   For example,

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

   For more details on creating a service group refer to,*Veritas Cluster Server User's Guide*

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

   For example,

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

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

   For example,

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

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

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

4. Create links between the resources. For example,

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

5 Create SAPWebAS71 resource for SAP. For example,

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

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

See "SAP WebAS agent attributes" on page 31.

6 Create resource dependencies for SAP WebAS resource.

The SAPWebAS71 resource depends on IP and Mount resources.

For example,

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

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

```
# hares -dep

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

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 a SAP instance.
- The CCMS agent, in case of any server on which the agent is installed and active.

Functional principle of CCMS Agents

The CCMS agents process the following tasks simultaneously:

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

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

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

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

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

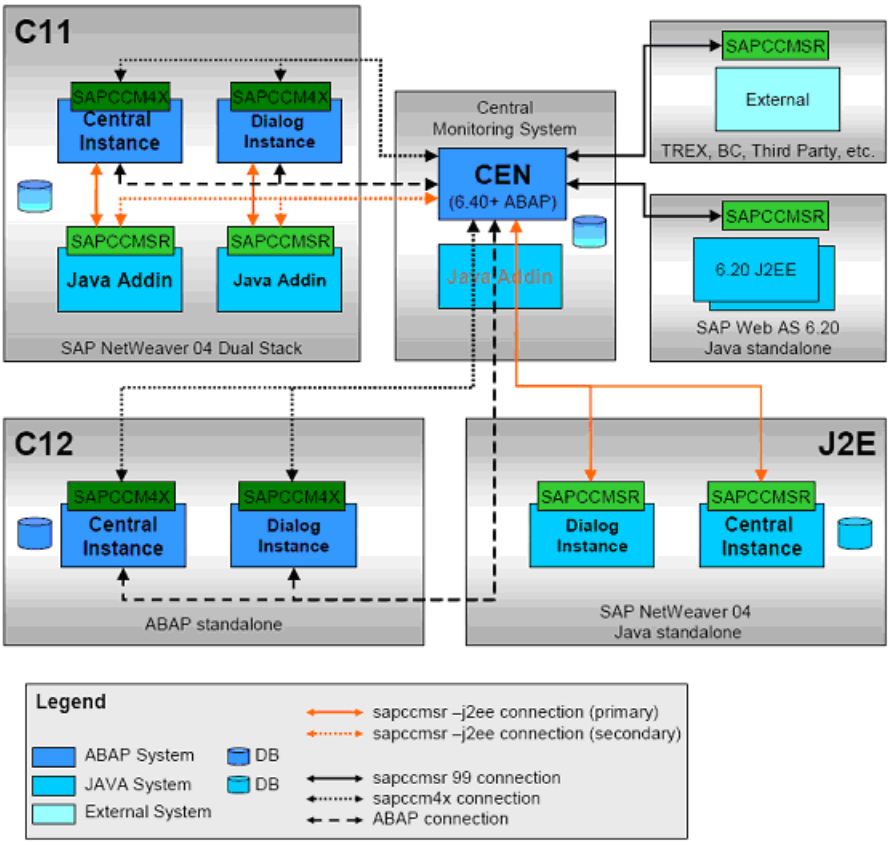
As an RFC server, it provides access to the data in the monitoring segment. For example, you can access this data using transaction RZ20. The agent then automatically creates the local configuration file and the RFC destination in the central system during its registration.

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

Possible Central Monitoring Scenarios with CCMS Agents

Figure 4-1 shows the central monitoring scenarios possible with different SAP NetWeaver components like ABAP, Java and Add-In (dual stack).

Figure 4-1 Central monitoring scenarios possible with different SAP WebAS 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 Sytem > 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 Veritas agent for SAP WebAS

The Veritas agent for SAP WebAS 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 Veritas agent for SAP WebAS.

To configure the CCMS agent with the Veritas agent for SAP WebAS

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 SAP server instances for cluster support

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

Synchronizing accounts and services

Synchronize user and group accounts 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 HA environment directly using virtual hostnames. To install SAP using virtual hostname, perform the following steps:

Note: Before installing SAP system refer to SAP Installation documentation.

To install SAP using Virtual Hostname

1. Go to the directory where sapinst tool is present in the Master DVD.
2. Launch the SAPInst GUI using the following command:

   ```
   sapinst SAPINST_USE_HOSTNAME=VirtualHostName
   ```

3. From 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.

Configuring SAPWebAS71 preonline script

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

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

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

The SAPWebAS71 preonline script performs the following tasks:

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

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

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

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

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

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

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

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

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

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

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

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

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

```
my $sResLogLevel = 'TRACE'; # Define logging level..
my @lsCmdArgs = ( @ARGV, $sResLogLevel ); # Insert logging level..
my $sArgs = join ( ' ', @lsCmdArgs);
my $iExitCode = undef;
#----------------------------------------
# Pass control to preonline, if it exists..
#----------------------------------------
if ( -x $sCmd ) {
  VCSAG_LOG_MSG ("I", "Preonline Cmd [$sCmd]
  Args [$sArgs]", 15031);
```

```
  system ( $sCmd, @lsCmdArgs );
#-------------------
# Exit if successful..
#----------------------------
  exit $iExitCode unless ( $iExitCode = $?>> 8 );
}
# give control back to HAD.
if (defined $ARGV[3]) {
   system("$vcs_home/bin/hagrp -online -nopre $ARGV[1] -sys
 $ARGV[0] -checkpartial $ARGV[3]");
   exit;
}
system("$vcs_home/bin/hagrp -online -nopre $ARGV[1]
-sys $ARGV[0]");
exit;
```

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

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

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

- Copy this file in the $VCS_HOME/bin/triggers directory.
- Ensure that the file is executable, and accessible to the "root" user.

5 For the service group, set the preonline flag to True.

```
hagrp -modify service_group PreOnline 1
```

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 as follows:

hagrp -modify service_group PreOnline 0

Clustering shared file systems

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

For other application servers

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

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

Configuring the 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.

Chapter 5

Troubleshooting the agent for SAP WebAS

This chapter includes the following topics:

Using correct software and operating system versions

Ensure that no issues arise due to incorrect software and operating system versions. For the correct versions of operating system and software to be installed on the resource systems:

See "Supported software" on page 12.

Meeting prerequisites

Before installing the agent for SAP WebAS, double check that you meet the prerequisites.

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

See "Before you install the Veritas agent for SAP WebAS" on page 21.

Configuring SAP Web AS resources

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

Starting the SAP Web AS instance outside a cluster

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

You can then restart the SAP Web AS 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 illustrated as follows.

To restart the SAP instance outside the cluster framework

1. Log in as superuser.
2. Ensure that the SAP database is up and running. Refer to the relevant database documentation or consult your database administrator for more information.
3. Use the SAPAdmin attribute to log in to the SAP server.

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

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

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

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

```
$ sapstart pf=InstProfile
```

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

For example, for a 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 error log files

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

Using SAP Web AS log files

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

Reviewing cluster log files

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

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

---

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

Reviewing agent log files

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

Using trace level logging

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

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

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

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

To localize ResLogLevel attribute for a resource

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

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

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

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

4. Note the time before you begin to operate the identified resource.
5. Test the identified resource. The function reproduces the problem that you are attempting to diagnose.
6. Note the time when the problem is reproduced.

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

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

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

You can also contact Symantec support for more help.

Using trace level logging for preonline trigger

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

See "Configuring SAPWebAS71 preonline script" on page 66.

To set the ResLogLevel attribute for preonline trigger

1 Go to the $VCS_HOME/bin/triggers directory.

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

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

3 Edit the value of the ResLogLevel attribute:

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

4 Save and close the preonline file.

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

Checks for an SAP Add-in Usage Types

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

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

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

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

Configuration checks for Solaris zones support

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

- Prerequisites for enabling Solaris zone support
  See "Before you install the Veritas agent for SAP WebAS" on page 21.
- Importing the types.cf file for Solaris zone support
  See "Importing the agent types files in a VCS environment" on page 30.
- Configuring the SAP resources for Solaris zone support
  See "Setting up zones on Solaris for SAP Enqueue and Enqueue Replication Servers" on page 40.

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.

For details on configuring the preonline script,

See "Configuring SAPWebAS71 preonline script" on page 66.

Alternatively, perform the following steps:

- Uninstall the agent package from the node.
  See "Removing the agent in a VCS environment" on page 24.
- Boot the zone on the node.
- Install the package on the node.
  See "Installing the agent in a VCS environment" on page 24.

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.

Appendix A

Sample Configurations

This appendix includes the following topics:

About sample configurations for the agent for SAP WebAS

The sample configuration graphically depicts the resource types, resources, and resource dependencies within the service group. Review these dependencies carefully before configuring the agent for SAP WebAS. For more information about these resource types, see the *Veritas Cluster Server Bundled Agents Reference Guide*.

Sample agent type definition for SAP WebAS

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

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

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

Sample SAP resource configuration

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

Sample SAP primary application server instance

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

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

Sample SAP additional application server instance

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

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

Sample SAP Central Services instance

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

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

Sample SAP Enqueue Replication server instance

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

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

Sample service group configuration for ABAP and Java Usage types

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

These characteristics include the following:

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

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

Figure A-1 Service group configuration for Application Server

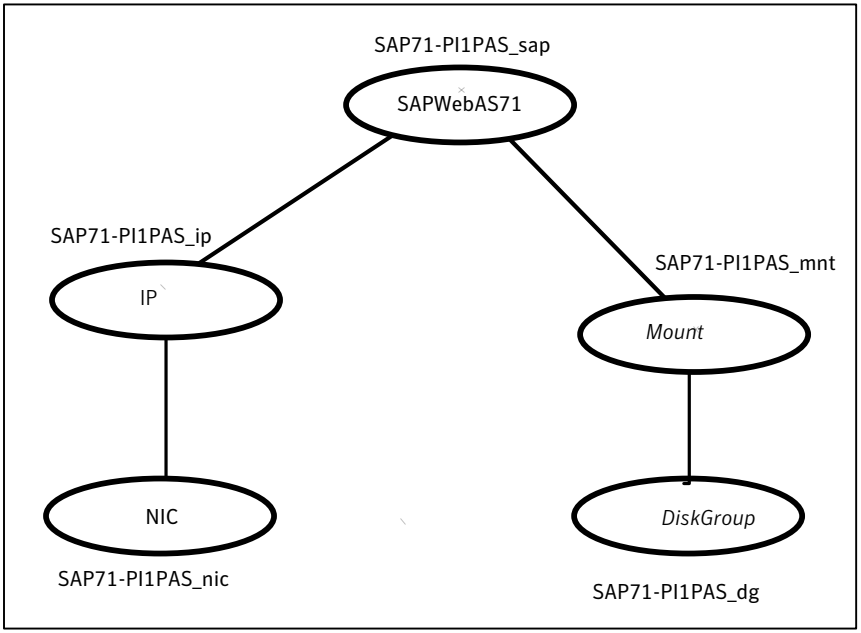


Figure A-2 shows a sample service group configuration for Enqueue Server instance.

Figure A-2 Service group configuration for Enqueue Server instance

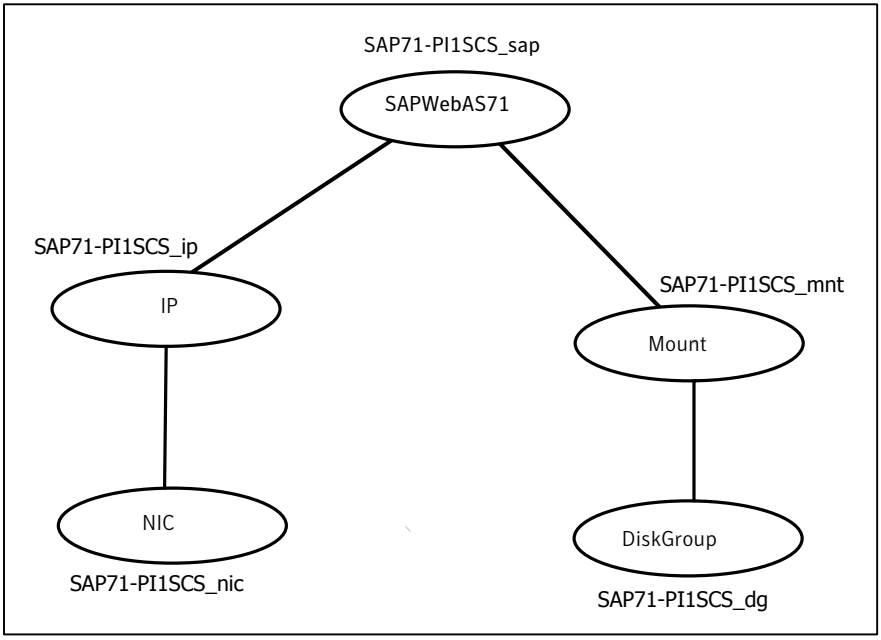
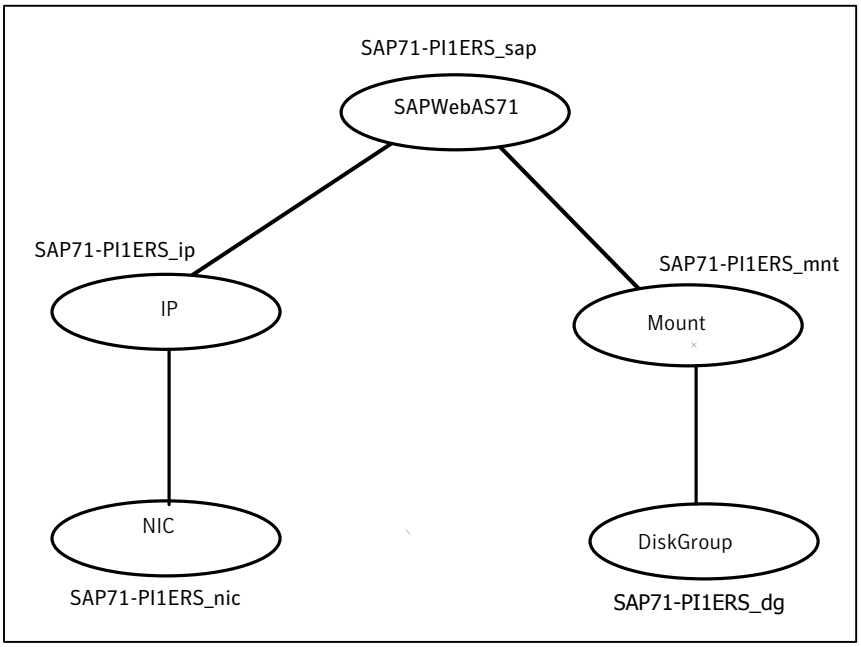


Figure A-3 shows a sample service group configuration for Enqueue Replication Server instance.

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

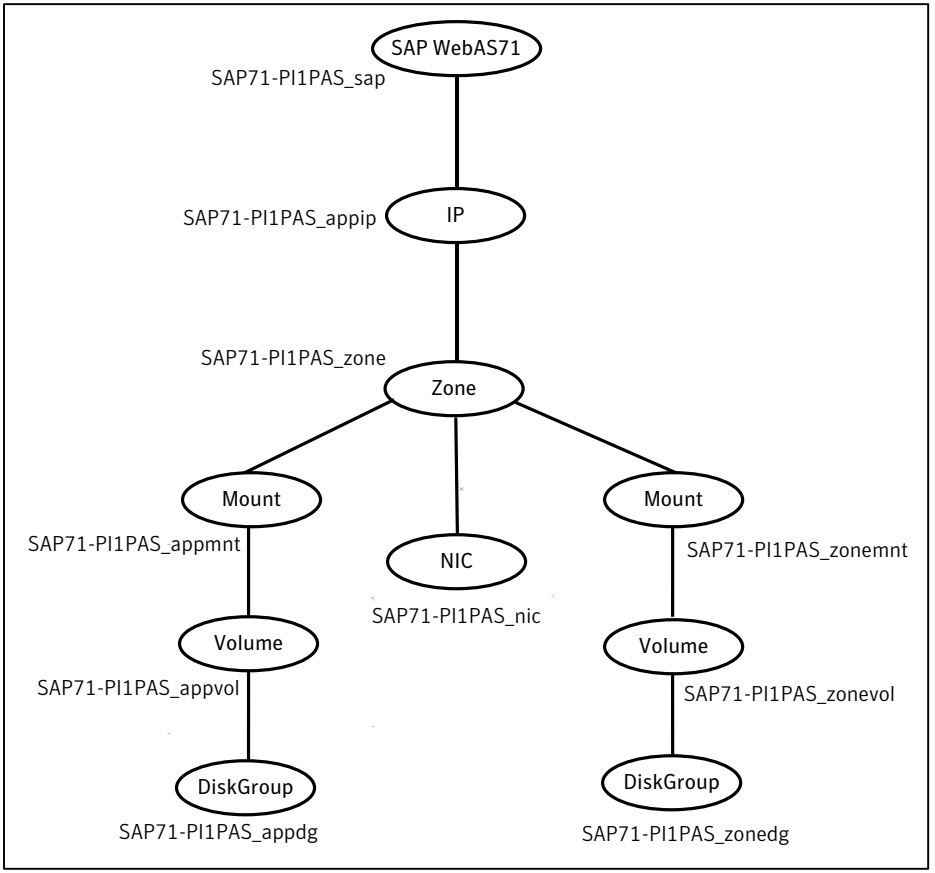


Sample SAP WebAS service group configurations for Solaris zone support

This section includes sample service groups with Solaris zone support.

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

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

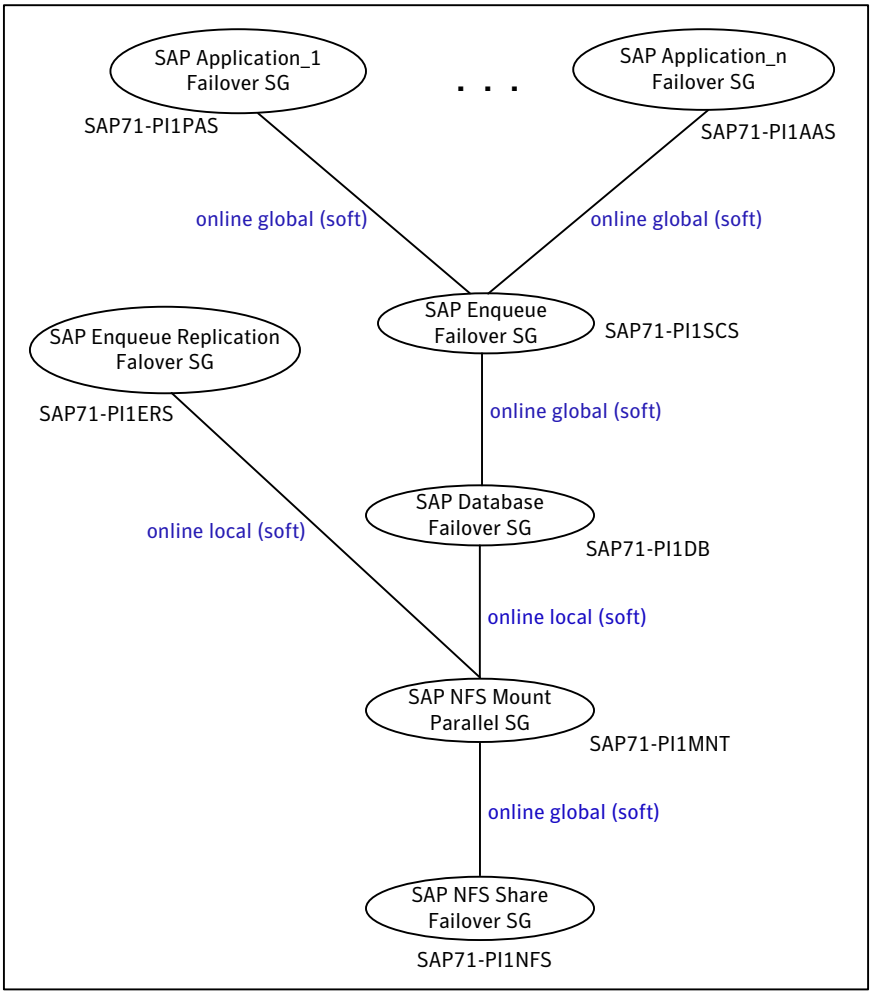


Sample service group dependency for SAP WebAS

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

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

Figure A-5 Sample service group dependency



Index

P

R

S

T

U