

Cluster Server Agent for SAPComponents Installation and Configuration Guide

AIX, HP-UX, Linux, Solaris

7.0

Last updated: 2021-01-07

Legal Notice

Copyright © 2021 Veritas Technologies LLC. All rights reserved.

Veritas and the Veritas Logo are trademarks or registered trademarks of Veritas Technologies LLC or its affiliates in the U.S. and other countries. Other names may be trademarks of their respective owners.

This product may contain third-party software for which Veritas is required to provide attribution to the third-party ("Third-Party Programs"). Some of the Third-Party Programs are available under open source or free software licenses. The License Agreement accompanying the Software does not alter any rights or obligations you may have under those open source or free software licenses. Refer to the third-party legal notices document accompanying this Veritas product or available at:

<https://www.veritas.com/about/legal/license-agreements>

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

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

The Licensed Software and Documentation are deemed to be commercial computer software as defined in FAR 12.212 and subject to restricted rights as defined in FAR Section 52.227-19 "Commercial Computer Software - Restricted Rights" and DFARS 227.7202, et seq. "Commercial Computer Software and Commercial Computer Software Documentation," as applicable, and any successor regulations, whether delivered by Veritas as on premises or hosted services. Any use, modification, reproduction release, performance, display or disclosure of the Licensed Software and Documentation by the U.S. Government shall be solely in accordance with the terms of this Agreement.

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

Technical Support

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

<https://www.veritas.com/support>

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

<https://my.veritas.com>

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

Worldwide (except Japan)

CustomerCare@veritas.com

Japan

CustomerCare_Japan@veritas.com

Documentation

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

<https://sort.veritas.com/documents>

Documentation feedback

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

infoscaledocs@veritas.com

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

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

Veritas Services and Operations Readiness Tools (SORT)

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

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

Contents

Chapter 1	Introducing the agent for SAPComponents	6
	About the Cluster Server agent for SAPComponents	6
	Supported software	7
	How the agent makes SAPComponents highly available	7
	Agent functions	7
	Online	7
	Offline	8
	Monitor	9
	Clean	9
 Chapter 2	 Installing and configuring SAPComponents for high availability	 10
	About SAPComponents	10
	SAP Web Dispatcher	11
	SAP Gateway	13
	SAP Master Data Management	14
	SAP Content Server	16
	Uniquely identifying an SAPComponents instance	17
	Monitoring an SAPComponents instance	18
	About configuring SAPComponents for high availability	18
	Configuring SAPComponents instances for cluster support	18
	Synchronizing accounts and services	19
	Installing SAPComponents using virtual hostname	19
 Chapter 3	 Installing, upgrading, and removing the agent for SAPComponents	 20
	Before you install the Cluster Server agent for SAPComponents	20
	About the ACC library	21
	Installing the ACC library	21
	Installing the ACC library IPS package on Oracle Solaris 11 systems	22
	Installing the agent in a VCS environment	23
	Installing the agent IPS package on Oracle Solaris 11 systems	25
	Uninstalling the agent in a VCS environment	25

	Removing the ACC library	26
	Upgrading the agent in a VCS environment	27
Chapter 4	Configuring the agent for SAPComponents	30
	About configuring the Cluster Server agent for SAPComponents	30
	Importing the agent types files in a VCS environment	30
	Agent attributes	32
	Specifying a custom monitor utility	35
Chapter 5	Configuring service groups for SAPComponents using the command line	37
	About configuring service groups for SAPComponents	37
	Before configuring the service groups for SAPComponents	38
	Configuring service groups for SAPComponents	38
	Generating the environments file for SAPComponents	42
Chapter 6	Troubleshooting the agent for SAPComponents	43
	Using the correct software and operating system versions	43
	Meeting prerequisites	43
	Configuring SAPComponents resources	44
	Starting the SAPComponents instance outside a cluster	44
	Reviewing error log files	45
	Using SAPComponents log files	45
	Reviewing cluster log files	45
	Using trace level logging	45
Appendix A	Sample Configurations	48
	About sample configurations for the agents for SAPComponents	48
	Sample agent type definition for SAPComponents	48
	VCS 4.x	49
	VCS 5.0 or later	49
	Sample SAP resource configuration	49
	Sample SAP Gateway instance	50
	Sample SAP Web Dispatcher instance	50
	Sample SAP Master Data Management instance	50
	Sample service group configuration for SAPComponents	51

Introducing the agent for SAPComponents

This chapter includes the following topics:

- [About the Cluster Server agent for SAPComponents](#)
- [Supported software](#)
- [How the agent makes SAPComponents highly available](#)
- [Agent functions](#)

About the Cluster Server agent for SAPComponents

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

The Cluster Server agent for SAPComponents provides high availability for the different SAP components in a cluster. The agent brings specific SAPComponents instances online, monitors the instances, and takes the instances offline. The agent monitors the processes of an SAPComponents instance and shuts down the SAPComponents in case of a failure.

The Cluster Server agent for SAPComponents supports the following standalone engines of an SAP system:

- SAP Gateway
- SAP Web Dispatcher
- SAP Master Data Management (MDM)

Supported software

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

How the agent makes SAPComponents highly available

The agent for SAPComponents continuously monitors the SAPComponents processes to verify that they function properly.

The agent provides the following levels of application monitoring:

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

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 operation performs the following tasks:

- Verifies that the required attributes are set correctly.
- Performs a preliminary check to ensure that the SAPComponents instance is not online on the specified node in the cluster. If the instance is online, the online operation exits immediately.
- Removes any SAPComponents processes that remain because of an unclean shutdown as follows:

- Kills the existing processes for SAPComponents instance.
- Executes the `cleanipc` utility for the SAP-provided SAPComponents instance.
- 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.
- Starts the SAPComponents instance using the `sapcontrol` commands. If the `sapcontrol` binary is not found, the `startsap` command is used. The online function also sources a shell script or a program that the `EnvFile` attribute specifies.
The command always gets executed in the context of SAPAdmin.
- Ensures that the SAPComponents instance is up and running successfully. The operation uses the wait period that the `OnlineTimeout` attribute specifies, to enable the SAPComponents instance to initialize fully before allowing the monitor function to probe the resource.

Offline

The offline operation performs the following tasks:

- Verifies that the required attributes are set correctly.
- Checks if the SAPComponents instance is already offline.
- If the instance is already offline, the operation verifies if any processes belonging to this SAPComponents instance exist.
- Stops the SAPComponents instance using the `sapcontrol` commands. If the `sapcontrol` binary is not found, the `stopsap` command is used. The offline function also sources a shell script or a program that the `EnvFile` attribute specifies.
The command always gets executed in the context of SAPAdmin.
- Ensures that no relevant SAPComponents processes are running. If any processes remain, the operation kills the remaining processes using a SIGKILL signal.
- Removes any existing IPC resources using the SAP-provided `cleanipc` utility.
- If the `kill.sap` and `shutdown.sap` files exist in the `/usr/sap/SAPSID/InstName/work` directory, the function removes the file from the directory.

Monitor

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

- The first-level check scans the system process table and searches for the processes that must be running for the SAPComponents instance. If the first-level check does not find these processes running on the node, the check exits immediately, and reports the SAPComponents instance as offline.
- If the `SecondLevelMonitor` attribute is greater than 0, the monitor function performs a thorough health check of the SAPComponents instance using the `sapcontrol` command.
- Depending upon the `MonitorProgram` attribute, the monitor function can perform a customized check using a user-supplied monitoring utility. For details about executing a custom monitor program: See [“Specifying a custom monitor utility”](#) on page 35.

Clean

In case of a failure or after an unsuccessful attempt to bring SAPComponents online or take SAPComponents offline, the clean function removes any SAPComponents processes remaining on the system. The function performs the following tasks:

- Attempts to gracefully shut down the SAPComponents instance using the `sapcontrol` command.
This command always gets executed in the context of SAPAdmin.
- Ensures that no relevant SAPComponents processes are running. If any processes remain, the operation kills all the remaining processes using a SIGKILL signal.
- Removes any existing IPC resources using the SAP-provided `cleanipc` utility.
- If the `kill.sap` and `shutdown.sap` files exist in the `/usr/sap/SAPSID/InstName/work` directory, the function removes the file from the directory.

Installing and configuring SAPComponents for high availability

This chapter includes the following topics:

- [About SAPComponents](#)
- [Uniquely identifying an SAPComponents instance](#)
- [Monitoring an SAPComponents instance](#)
- [About configuring SAPComponents for high availability](#)
- [Configuring SAPComponents instances for cluster support](#)

About SAPComponents

SAPComponents includes the following standalone engines of an SAP system:

- SAP Web Dispatcher
- SAP Gateway
- SAP Master Data Management (MDM)
- SAP Content Server

Note: The information in the following sections is based on the SAP Web Dispatcher, SAP Gateway, and SAP MDM product documentation available on the SAP Help Portal site.

SAP Web Dispatcher

SAP Web Dispatcher is a program that is designed to relay incoming HTTP requests to your SAP system, and consequently, balance the load in your SAP system. When SAP Web Dispatcher accepts a connection, it balances the load to ensure an even distribution across the servers.

You can use SAP Web Dispatcher in ABAP/Java systems and in pure Java systems, as well as in pure ABAP systems.

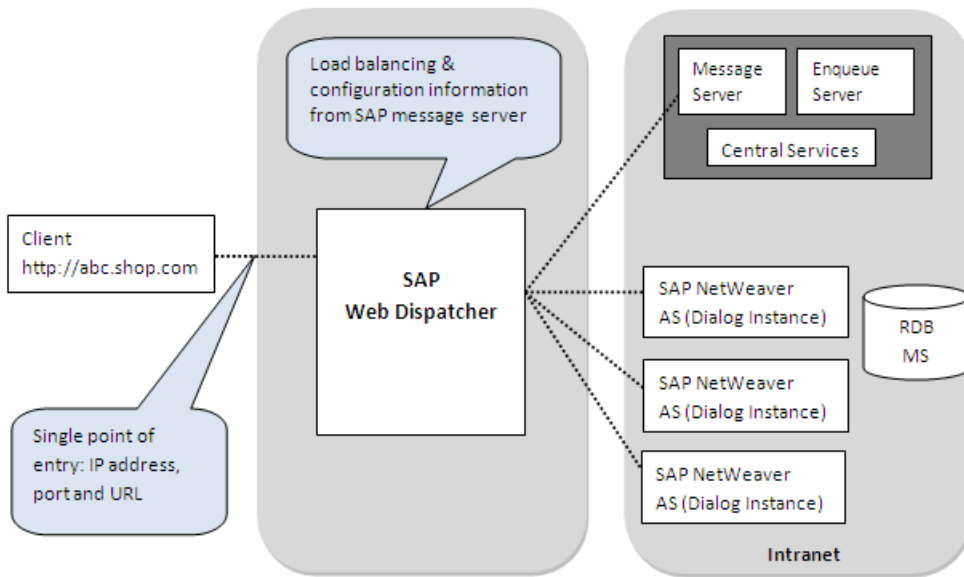
SAP recommends SAP Web Dispatcher when you use an SAP system with several SAP NetWeaver application server instances for Web applications.

SAP Web Dispatcher performs the following tasks:

- Selects the appropriate application server.
- Accepts or rejects HTTP requests, depending on the URL. By defining the URLs that you want to be rejected, you can restrict access to your system.
- Serves as a Web Cache to improve the response times and to conserve the application server cache.

SAP Web Dispatcher architecture

You can run SAP Web Dispatcher on the machine that is connected directly to the Internet. SAP Web Dispatcher is located between the Web client (browser) and your SAP system that is running the Web application.

Figure 2-1 SAP Web Dispatcher architecture

On accepting a connection, SAP Web Dispatcher forwards incoming HTTP and HTTPS requests to the appropriate application server (AS) of the SAP system. SAP Web Dispatcher selects the appropriate AS based on the following:

- The capacity of the AS. The capacity of an AS ABAP depends on the number of configured dialog work processes. The capacity of an AS Java depends on the number of Server Processes.
- The state of the application.
- SAP Web Dispatcher decides whether the incoming request should be forwarded to an ABAP or Java server.

SAP Web Dispatcher has some architectural similarities to Internet Communication Manager (ICM). However, unlike the ICM, SAP Web Dispatcher passes incoming requests to the ICM of the application server. The response of the application server returns to the client using the same network connection via SAP Web Dispatcher.

If the AS acting as the client opens connections to external HTTP servers, these connections do not go via SAP Web Dispatcher; these connections go directly to the server (or possibly via a configured proxy).

SAP Gateway

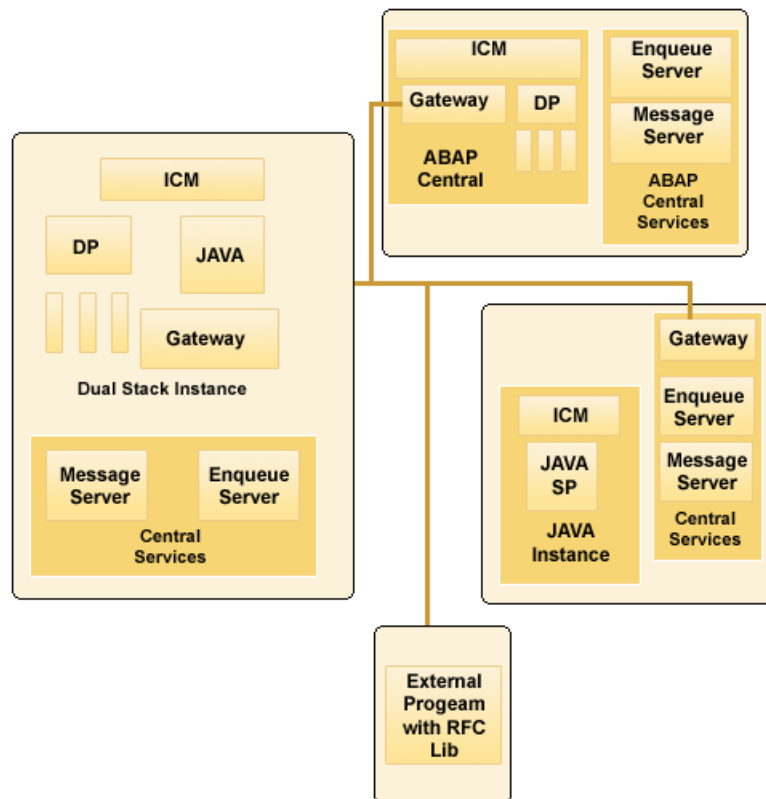
Every instance of an SAP system has a gateway. SAP Gateway carries out TCP/IP-based RFC services within the SAP world. These services enable communication between:

- SAP Systems and external programs
- Processes of an instance or a system, or between systems

SAP Gateway architecture

Figure 2-2 shows three different SAP systems that use RFC connections via gateways.

Figure 2-2 SAP Gateway architecture



- For systems that contain pure ABAP or dual-stack AS, each instance contains a gateway that ABAP Dispatcher starts and monitors.

If a system is installed with ASCS instance (ABAP Central Services, made up of the message server and standalone enqueue server), the system does not contain a gateway.

- In a Java-only system, the instances communicate with each other without using the RFC. The gateway is used for RFC/JCo connections to other systems.

For certain implementations, an SAP Gateway is used as a separately-installed SAP instance or used outside an SAP system.

SAP Master Data Management

The SAP Master Data Management (MDM) module is built on top of SAP XI/PI Netweaver, which is used to consolidate master data objects from multiple different SAP remote systems.

SAP MDM follows the consolidated model of collecting master data from several systems and to store the details in a central business partner repository. Consolidation allows for detection and a cleansing of duplicate and identical records.

SAP MDM supports the following master data objects:

- Customers
- Vendors
- Employees
- Materials
- Products
- Business Partners

SAP MDM Architecture

[Figure 2-3](#) represents an architectural diagram of a typical deployment when using the MDM ABAP API.

Figure 2-3 Typical deployment - MDM ABAP API

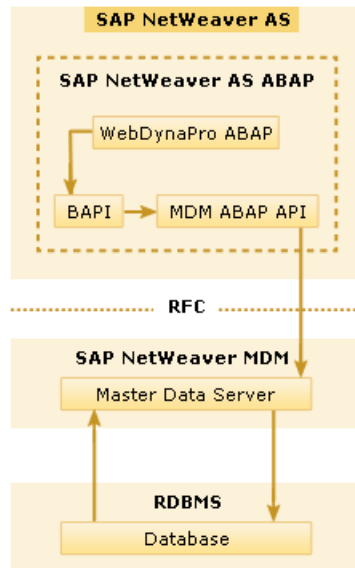
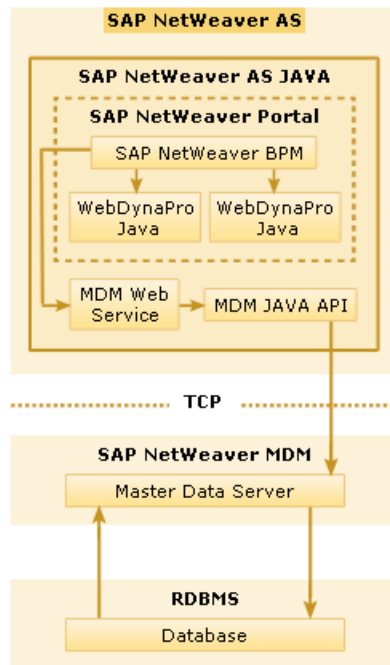


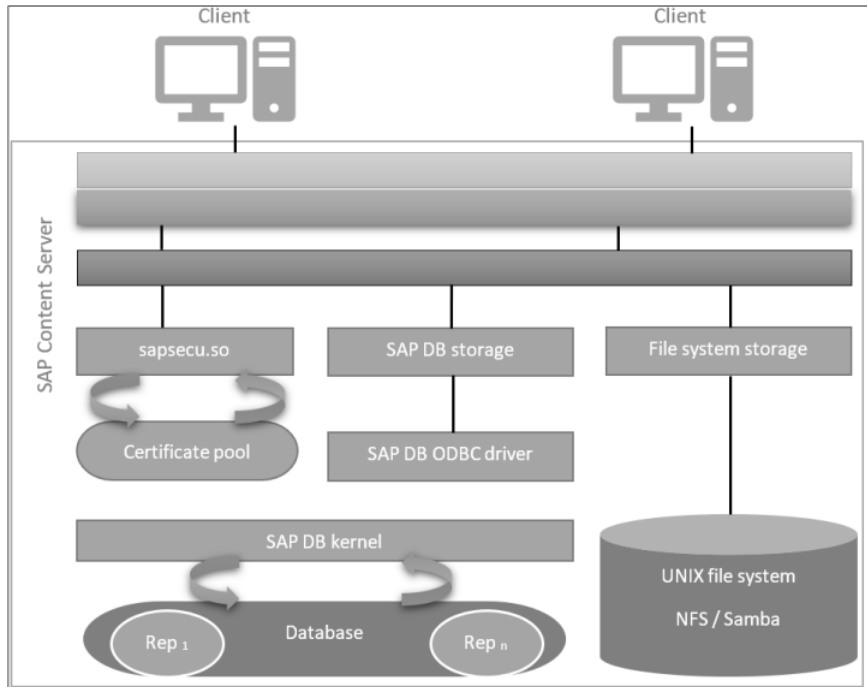
Figure 2-4 represents an architectural diagram of a typical deployment when using the MDM Java API.

Figure 2-4 Typical deployment - MDM Java API

SAP Content Server

SAP Content Server provides the technical infrastructure for all document-centric applications and business scenarios that do not require a long-term document archiving solution. A self-contained content server is always available to SAP customers, because SAP Content Server is included in every SAP solution.

SAP 7.5 onwards, Content Server is integrated into SAP Web Dispatcher instead of the Apache (UNIX) or the Microsoft IIS (Windows) web servers. SAP Content Server is typically installed under its own system ID (SAPSID), and it has an instance number that is unique within the system. Thus, SAP Content Server instances are better integrated in your SAP system landscape.

Figure 2-5 SAP Content Server

Uniquely identifying an SAPComponents instance

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

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

Each instance has a unique instance name. The instance names may follow the conventional form. For example, gateway instances begin with 'G', and Web Dispatcher instances typically begin with 'W'. Instance names often include an instance ID suffix, which is an integer between 00-99. For example, a Gateway instance with an instance ID = 00 may have an instance name of G00.

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

Differentiating SAPComponents 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 SAPComponents instance during a clean operation.

Monitoring an SAPComponents instance

- In first-level monitoring, the SAPComponents quickly checks for the existence of the processes for the SAPComponents instance. To do this, the agent scans the system process table and searches for strings in the process command line that uniquely identify the processes associated with the SAPComponents instance. These search strings are based on the value specified in the `InstProfile` and `ComponentType` resource attributes.
- In second-level monitoring, the agent runs the SAP-provided `sapcontrol` utility to verify the status of running process for the SAPComponents instance. On the basis of exit code and output of the utility, the agent decides the status of the SAPComponents instance.

About configuring SAPComponents for high availability

The guidelines for configuring SAPComponents for high availability are as follows:

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

Configuring SAPComponents instances for cluster support

This section describes pointers to configure an SAPComponents 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 SAPComponents using virtual hostname

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

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

To install SAPComponents using virtual hostname

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

```
sapinst SAPINST_USE_HOSTNAME=VirtualHostName
```

- 3 From the installation GUI, select **Standalone Engines > Gateway > Gateway Installation or Standalone Engines > Web Dispatcher > Web Dispatcher** based on the standalone engine you are planning to install.

Installing, upgrading, and removing the agent for SAPComponents

This chapter includes the following topics:

- [Before you install the Cluster Server agent for SAPComponents](#)
- [About the ACC library](#)
- [Installing the ACC library](#)
- [Installing the ACC library IPS package on Oracle Solaris 11 systems](#)
- [Installing the agent in a VCS environment](#)
- [Installing the agent IPS package on Oracle Solaris 11 systems](#)
- [Uninstalling the agent in a VCS environment](#)
- [Removing the ACC library](#)
- [Upgrading the agent in a VCS environment](#)

Before you install the Cluster Server agent for SAPComponents

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

Before you install the agent for SAPComponents, ensure that the following prerequisites are met.

- Install and configure Cluster Server.
For more information on installing and configuring Cluster Server, refer to the Cluster Server installation and configuration guides.
- Install the latest version of ACC Library.
To install or update the ACC Library package, locate the library and related documentation in the Agent Pack tarball.
See “[About the ACC library](#)” on page 21.

About the ACC library

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

Instructions to install or remove the ACC library on a single system in the cluster are given in the following sections. The instructions assume that the ACCLib tar file has already been extracted.

Note: The LogDbg attribute should be used to enable debug logs for the ACCLib-based agents when the ACCLib version is 6.2.0.0 or later and VCS version is 6.2 or later.

Installing the ACC library

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

To install the ACC library

- 1 Log in as a superuser.
- 2 Download ACC Library.

You can download either the complete Agent Pack tar file or the individual ACCLib tar file from the Veritas Services and Operations Readiness Tools (SORT) site (<https://sort.veritas.com/agents>).

- 3 If you downloaded the complete Agent Pack tar file, navigate to the directory containing the package for the platform running in your environment.

AIX	<code>cd1/aix/vcs/application/acc_library/version_library/pkgs</code>
HP-UX	<code>cd1/hpux/generic/vcs/application/acc_library/version_library/pkgs</code>
Linux	<code>cd1/linux/generic/vcs/application/acc_library/version_library/rpms</code>
Solaris	<code>cd1/solaris/dist_arch/vcs/application/acc_library/version_library/pkgs</code>

- 4 If you downloaded the individual ACCLib tar file, navigate to the pkgs directory (for AIX and Solaris), or rpms directory (for Linux).
- 5 Install the package. Enter **Yes**, if asked to confirm overwriting of files in the existing package.

AIX	<code># installp -ac -d VRTSaclib.bff VRTSaclib</code>
HP-UX	<code># swinstall -s 'pwd' VRTSaclib</code>
Linux	<code># rpm -i \ VRTSaclib-VersionNumber-GA_GENERIC.noarch.rpm</code>
Solaris	<code># pkgadd -d VRTSaclib.pkg</code>

See [“Installing the ACC library IPS package on Oracle Solaris 11 systems”](#) on page 22.

Note: The LogDbg attribute should be used to enable debug logs for the ACCLib-based agents when the ACCLib version is 6.2.0.0 or later and VCS version is 6.2 or later.

Installing the ACC library IPS package on Oracle Solaris 11 systems

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

Note: If the ACCLib package is already installed, you do not need to install the ACCLib .p5p package.

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

- 1 Copy the `VRTSacclib.p5p` package from the `pkgs` directory to the system in the `/tmp/install` directory.
- 2 Disable the publishers that are not reachable as package install may fail, if any, of the already added repositories are unreachable.

```
# pkg set-publisher --disable <publisher name>
```
- 3 Add a file-based repository in the system.

```
# pkg set-publisher -g /tmp/install/VRTSacclib.p5p Veritas
```
- 4 Install the package.

```
# pkg install --accept VRTSacclib
```
- 5 Remove the publisher from the system.

```
# pkg unset-publisher Veritas
```
- 6 Enable the publishers that were disabled earlier.

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

Installing the agent in a VCS environment

Install the agent for SAPComponents on each node in the cluster.

To install the agent in a VCS environment

- 1 Download the agent from the Veritas Services and Operations Readiness Tools (SORT) site: <https://sort.veritas.com/agents>.

You can download either the complete Agent Pack tar file or an individual agent tar file.
- 2 Uncompress the file to a temporary location, say `/tmp`.

- 3 If you downloaded the complete Agent Pack tar file, navigate to the directory containing the package for the platform running in your environment.

AIX	<code>cd1/aix/vcs/application/sap_agent/ vcs_version/version_agent/pkg</code>
HP-UX	<code>cd1/hpux/generic/vcs/application/sap_agent/ vcs_version/version_agent/pkg</code>
Linux	<code>cd1/linux/generic/vcs/application/sap_agent/ vcs_version/version_agent/rpms</code>
Solaris	<code>cd1/solaris/dist_arch/vcs/application/sap_agent/ vcs_version/version_agent/pkg</code>

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

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

AIX	<code># installp -ac -d VRTSsapcms.rte.bff VRTSsapcms.rte</code>
HP-UX	<code># swinstall -s `pwd` VRTSsapcms</code>
Linux	<code># rpm -ihv \ VRTSsapcms-AgentVersion-GA_GENERIC.noarch.rpm</code>
Solaris	<code># pkgadd -d . VRTSsapcms</code>

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

See [“Importing the agent types files in a VCS environment”](#) on page 30.

Installing the agent IPS package on Oracle Solaris 11 systems

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

- 1 Copy the `VRTSsapcms.p5p` package from the `pkgs` directory to the system in the `/tmp/install` directory.
- 2 Disable the publishers that are not reachable as package install may fail, if any of the already added repositories are unreachable.

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

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

- 3 Add a file-based repository in the system.

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

- 4 Install the package.

```
# pkg install --accept VRTSsapcms
```

- 5 Remove the publisher from the system.

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

- 6 Enable the publishers that were disabled earlier.

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

Uninstalling the agent in a VCS environment

You must uninstall the agent for SAPComponents 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 running the following command from any node in the cluster:

```
# haconf -makerw
```

- 3 Remove all SAPComponents resources from the cluster. Run the following command to verify that all resources have been removed:

```
# hares -list Type=SAPComponents
```

- 4 Remove the agent type from the cluster configuration by running the following command from any node in the cluster:

```
# hatype -delete SAPComponents
```

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 running 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 SAPComponents from each node in the cluster.

Run the following command to uninstall the agent:

```
AIX          # installp -u VRTSsapcms.rte
```

```
HP-UX        # swremove VRTSsapcms
```

```
Linux        # rpm -e VRTSsapcms
```

```
Solaris      # pkgrm VRTSsapcms
```

Note: To uninstall the agent IPS package on a Solaris 11 system, run the following command:

```
# pkg uninstall VRTSsapcms
```

Removing the ACC library

Perform the following steps to remove the ACC library.

To remove the ACC library

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

```
AIX          # installp -u VRTSaccLib
```

```
HP-UX        # swremove VRTSaccLib
```

```
Linux        # rpm -e VRTSaccLib
```

```
Solaris      # pkgrm VRTSaccLib
```

Note: To uninstall the ACCLib IPS package on a Solaris 11 system, run the following command:

```
# pkg uninstall VRTSaccLib
```

Upgrading the agent in a VCS environment

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

To upgrade the agent in a VCS environment

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

```
# hagr -freeze groupName -persistent
```

- 2 Stop the cluster services forcibly.

```
# hstop -all -force
```

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

```
# ps -ef | grep SAPComponents
```

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

Run the following command to uninstall the agent:

AIX # installp -u VRTSsapcms.rte

HP-UX # swremove VRTSsapcms

Linux # rpm -e VRTSsapcms

Solaris For Solaris 10:

 # pkgrm VRTSsapcms

 For Solaris 11:

 # pkg uninstall VRTSsapcms

- 5 Install the new agent on all the nodes.

See [“Installing the agent in a VCS environment”](#) on page 23.

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

VCS 4.x	■ AIX	/etc/VRTSvcs/conf/sample_SAPComponents/
	■ HP-UX	SAPComponentsTypes.cf
	■ Linux	
	■ Solaris	

VCS 5.x or later	■ AIX	/etc/VRTSagents/ha/conf/SAPComponents/
	■ HP-UX	SAPComponentsTypes.cf
	■ Linux	

VCS 5.0	■ Solaris SPARC	/etc/VRTSagents/ha/conf/SAPComponents/ SAPComponentsTypes50.cf
---------	--------------------	---

VCS 5.1 or later	■ Solaris SPARC	/etc/VRTSagents/ha/conf/SAPComponents/ SAPComponentsTypes51.cf
------------------	--------------------	---

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

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

- 8 Start the cluster services.

```
# hstart
```

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

```
# haagent -start SAPComponents -sys SystemName
```

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

```
# hagr -unfreeze GroupName -persistent
```

Configuring the agent for SAPComponents

This chapter includes the following topics:

- [About configuring the Cluster Server agent for SAPComponents](#)
- [Importing the agent types files in a VCS environment](#)
- [Agent attributes](#)
- [Specifying a custom monitor utility](#)

About configuring the Cluster Server agent for SAPComponents

After installing the Cluster Server agent for SAPComponents, you must import the agent type configuration file. After importing this file, review the attributes table that describes the resource type and its attributes, and then create and configure SAPComponents resources.

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

See [“About sample configurations for the agents for SAPComponents”](#) on page 48.

Importing the agent types files in a VCS environment

To use the agent for SAPComponents, you must import the agent types file into the cluster. You can import the agent types file using the VCS graphical user interface or using the command line interface.

To import the agent types file using the VCS Java GUI

- 1** Start the Cluster Manager (Java Console) and connect to the cluster on which the agent is installed.
- 2** Click **File > Import Types**.
- 3** In the **Import Types** dialog box, select the following file:

VCS 4.x	■ AIX	/etc/VRTSvcs/conf/sample_SAPComponents/
	■ HP-UX	SAPComponentsTypes.cf
	■ Linux	
	■ Solaris	
VCS 5.x or later	■ AIX	/etc/VRTSagents/ha/conf/SAPComponents/
	■ HP-UX	SAPComponentsTypes.cf
	■ Linux	
VCS 5.0	Solaris SPARC	/etc/VRTSagents/ha/conf/SAPComponents/ SAPComponentsTypes50.cf
VCS 5.1 or later	Solaris SPARC	/etc/VRTSagents/ha/conf/SAPComponents/ SAPComponentsTypes51.cf

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

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

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

To import the agent types file using the CLI

- 1** Log on to any one of the systems in the cluster as the superuser.
- 2** Create a temporary directory.

```
# mkdir ./temp
# cd ./temp
```

3 Copy the sample file `Types.cf`.

VCS 4.x	AIX	/etc/VRTSvcs/conf/sample_SAPComponents/
	HP-UX	SAPComponentsTypes.cf
	Linux	
	Solaris	
VCS 5.x or later	AIX	/etc/VRTSagents/ha/conf/SAPComponents/
	HP-UX	SAPComponentsTypes.cf
	Linux	
VCS 5.0	Solaris SPARC	/etc/VRTSagents/ha/conf/SAPComponents/
		SAPComponentsTypes50.cf
VCS 5.1 or later	Solaris SPARC	/etc/VRTSagents/ha/conf/SAPComponents/
		SAPComponentsTypes51.cf

4 Create a dummy `main.cf` file.

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

5 Create the SAP resource type as follows:

```
# hacf -verify .
# haconf -makerw
# sh main.cmd
# haconf -dump
```

The SAPComponents agent type is now imported to the VCS engine. You can now create SAPComponents resources. For additional information about using the VCS CLI, refer to the *Cluster Server Administrator's Guide*.

Agent attributes

[Table 4-1](#) shows the required attributes for configuring an SAPComponents Server instance:

Table 4-1 Required attributes

Required attributes	Description
ResLogLevel	<p>Specifies the logging detail that the agent performs for the resource.</p> <p>The valid values are as follows:</p> <ul style="list-style-type: none">■ ERROR: Only logs error messages■ WARN: Logs error messages and warning messages■ INFO: Logs error messages, warning messages, and informational messages■ TRACE: Logs error messages, warning messages, informational messages, and trace messages TRACE is very verbose and should be used only during initial configuration or for troubleshooting and diagnostic operations. <p>Type and dimension: string-scalar</p> <p>Default value: INFO</p> <p>Example: INFO</p>
SAPAdmin	<p>SAP system administrator for SAPComponents instance. This user name is usually a concatenation of the SAPSID and the adm string 'sidadm'.</p> <p>Storing SAPAdmin in system naming services is not supported, for example: NIS, NIS+ and LDAP servers.</p> <p>The agent functions use this user name to execute their respective core subroutines.</p> <p>Type and dimension: string-scalar</p> <p>Default value: No default value</p> <p>Example: gw1adm</p>

Table 4-1 Required attributes (*continued*)

Required attributes	Description
EnvFile	<p>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 SAPComponents scripts for online, offline, monitor, and clean operations.</p> <p>Supported shell environments are ksh, sh, and csh.</p> <p>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 the environments file for SAPComponents” on page 42.</p> <p>Veritas recommends that you store this file on shared disk so that the file is always available to an online system.</p> <p>Type and dimension: string-scalar</p> <p>Default value: No default value</p> <p>Example: <code>/usr/sap/GW1/G00/envfile</code></p>
InstProfile	<p>The full path to the SAP Instance profile.</p> <p>The Instance profile is found in the <code>/usr/sap/SAPSID/SYS/profile</code> 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.</p> <p>Type and dimension: string-scalar</p> <p>Default value: No default value</p> <p>Example: <code>/usr/sap/GW1/SYS/profile/GW1_G00_sapgwr</code></p>
ComponentType	<p>String identifier that classifies and describes the SAPComponents type. ComponentType values are case-sensitive. Valid values are:</p> <ul style="list-style-type: none"> ■ Gateway : SAP Gateway Server ■ Web Dispatcher: SAP Web Dispatcher ■ MDM: SAP MDM Component ■ Content Server WD: SAP Content Server Web Dispatcher <p>Type and dimension: string-scalar</p> <p>Default value: No default value</p> <p>Example: Gateway</p>

Table 4-2 shows the required attributes for configuring an SAPComponents Server instance:

Table 4-2 Optional attributes

SecondLevelMonitor	<p>Used to enable second-level monitoring. Second-level monitoring is a deeper, more thorough state check of the SAPComponents instance.</p> <p>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.</p> <p>Note: Exercise caution while setting SecondLevelMonitor to large numbers. For example, if the MonitorInterval is set to 60 seconds and the SecondLevelMonitor is set to 100, then sapcontrol is executed every 100 minutes, which may not be as often as intended. For maximum flexibility, no upper limit is defined for SecondLevelMonitor.</p> <p>Type and dimension: integer-scalar</p> <p>Default value: 0</p> <p>Example: 1</p>
MonitorProgram	<p>Absolute path name of an external, user-supplied monitor executable.</p> <p>For more information on setting this attribute: See “Specifying a custom monitor utility” on page 35.</p> <p>Default value: No default value</p> <p>Example 1: <code>\$ServerRoot/bin/myMonitor.pl</code></p> <p>Example 2: <code>\$ServerRoot/bin/myMonitor.sh arg1 arg2</code></p>

Specifying a custom monitor utility

The monitor function can execute a customized monitoring utility to perform an additional SAPComponents instance state check. The monitor function executes the utility specified in the MonitorProgram attribute if the following conditions are satisfied:

- The MonitorProgram attribute value is set to a valid executable utility.

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

The monitor operation interprets the program exit code as follows:

0	SAPComponents instance is online
110	SAPComponents instance is online
100	SAPComponents instance is offline
1	SAPComponents instance is offline
All other	SAPComponents instance is unknown

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

Configuring service groups for SAPComponents using the command line

This chapter includes the following topics:

- [About configuring service groups for SAPComponents](#)
- [Before configuring the service groups for SAPComponents](#)
- [Configuring service groups for SAPComponents](#)
- [Generating the environments file for SAPComponents](#)

About configuring service groups for SAPComponents

Configuring the SAPComponents service group involves creating the SAPComponents service group, its resources, and defining attribute values for the configured resources. You must have administrator privileges to create and configure a service group.

You can configure the service groups using one of the following:

- The Cluster Manager (Java console)
- Veritas Infoscale Operations Manager
- The command line

See [“Configuring service groups for SAPComponents”](#) on page 38.

Before configuring the service groups for SAPComponents

Before you configure the SAPComponents service group, you must:

- Verify that the Cluster Server components are installed and configured on all nodes in the cluster where you will configure the service group.
For more information on installing the components, refer to the *InfoScale Availability Installation Guide*.
- Verify that SAPComponents (SAP Gateway/SAP Web Dispatcher) is installed and configured identically on all the systems in the cluster.
- Verify that the Cluster Server agent for SAPComponents is installed on all nodes in the cluster.
- Verify that the type definition for the Cluster Server agent for SAPComponents is imported into the VCS engine.

See [“Importing the agent types files in a VCS environment”](#) on page 30.

Configuring service groups for SAPComponents

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

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

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

To add a service group for SAPComponents

- 1 Create a service group for SAPComponents. For example:

```
# hagrps -add SAP_Gateway
```

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

- 2 To add systems, modify the SystemList attribute for the group. For example:

```
# hagrps -modify SAP_Gateway SystemList systemA 0 systemB 1
```

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

```
# hares -add SAP_GW_nic NIC SAP_Gateway
```

```
# hares -add SAP_GW_ip IP SAP_Gateway
```

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

- 4 Create links between the resources. For example:

```
# hares -link SAP_GW_ip SAP_GW_nic
```

- 5 Create SAPComponents resource for the SAP component. In this case, the SAP component is SAP Gateway.

```
# hares -add SAP_GW_Res SAPComponents SAP_Gateway
```

Based on the SAPComponent instance you are clustering, modify the resource attributes.

See [“Agent attributes”](#) on page 32.

- 6** Create resource dependencies for the SAPComponents resource. The SAPComponents resource depends on the IP and Mount resources.

```
# hares -link SAP_GW_Res SAP_GW_ip
# hares -link SAP_GW_Res SAP_GW_mnt
```

- 7** Verify the final resource dependencies for the SAP_Gateway server group. For example,

```
# hares -dep
```

For example,

Group	Parent	Child
SAP_Gateway	SAP_GW_ip	SAP_GW_nic
SAP_Gateway	SAP_GW_mnt	SAP_GW_vol
SAP_Gateway	SAP_GW_res	SAP_GW_mnt
SAP_Gateway	SAP_GW_res	SAP_GW_ip
SAP_Gateway	SAP_GW_vol	SAP_GW_dg

The sample main.cf for SAPComponents service group is as follows:

```
include "types.cf"
include "SAPComponentsTypes.cf"
cluster cluster1 (
    UserNames = { admin = dlmElgLimHmKumGlj }
    ClusterAddress = "110.120.162.128"
    Administrators = { admin }
    UseFence = SCSI3
    HacliUserLevel = COMMANDROOT
)
system systemA (
)
system systemB (
)
system systemC (
)

group SAP_Gateway (
    SystemList = { systemA = 0, systemB =1 }
)
SAPComponents SAP_GW_Res1 (
    Critical = 0
    EnvFile = "/home/gwladm/.login"
    SAPAdmin = gwladm
```



```

    InstProfile = "/usr/sap/GW1/SYS/profile/GW1_W00_sapgw"
    ComponentType = Gateway
    SecondLevelMonitor = 1
    MonitorProgram = "/tmp/mymonitor.sh"
)
DiskGroup SAP_GW_dg (
    DiskGroup = sapgw1_dg
)
IP SAP_GW_ip (
    Device = bge0
    Address = "110.120.62.18"
    NetMask = "255.255.255.0"
)
Mount SAP_GW_mnt (
    MountPoint = "/usr/sap/GW1"
    BlockDevice = "/dev/vx/dsk/sapgw1_dg/sapgw1_vol"
    FSType = vxfs
    FsckOpt = "-y"
)
NIC SAP_GW_nic (
    Device = bge0
    Network Type = ether
)
SAP_GW_Res requires SAP_GW_ip
SAP_GW_Res requires SAP_GW_mnt
SAP_GW_ip requires SAP_GW_nic
SAP_GW_mnt requires SAP_GW_dg

```

```

// resource dependency tree
//
//      group SAP_Gateway
//      {
//          SAPComponents SAP_GW_Res
//          {
//              IP SAP_GW_ip
//              {
//                  NIC SAP_GW_nic
//              }
//              Mount SAP_GW_mnt
//              {
//                  DiskGroup SAP_GW_dg
//              }
//          }
//      }

```

```
//      }  
//      }
```

Generating the environments file for SAPComponents

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

To generate the environments file for SAPComponents

- 1 Log in as SAPAdmin user.

```
su - gwladm
```

- 2 Capture the environment with the following command.

```
env > /home/gwladm/sapgwlenv.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 `sapgwlenv.env` file to add string 'setenv' at the beginning of each line.
- Replace the '=' with space " " in the file.

- 4 Copy the `sapgwlenv.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 sapgwlenv.env
```

Troubleshooting the agent for SAPComponents

This chapter includes the following topics:

- [Using the correct software and operating system versions](#)
- [Meeting prerequisites](#)
- [Configuring SAPComponents resources](#)
- [Starting the SAPComponents instance outside a cluster](#)
- [Reviewing error log files](#)

Using the correct software and operating system versions

Ensure that you use correct software and operating system versions.

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

Meeting prerequisites

Before installing the agent for SAPComponents, ensure that the following prerequisites are met.

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

See [“Before you install the Cluster Server agent for SAPComponents”](#) on page 20.

Configuring SAPComponents resources

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

Starting the SAPComponents 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 SAPComponents instance independent of the cluster framework.

You can then restart the SAPComponents instance outside the cluster framework.

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

A sample procedure to start an SAPComponents instance outside the cluster framework is illustrated as follows.

To restart the SAPComponents instance outside the 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 command:

```
startsap r3 InstName SAPHost
```

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

For example, for an SAP instance:

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

If the SAPComponents 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 SAPComponents or the agent for SAPComponents, use the log files described in this section to investigate the problems.

Using SAPComponents log files

If the SAPComponents instance is facing problems, you can access the SAP 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 SAPComponents, 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 SAPComponents agent log files located at `/var/VRTSvcs/log/SAPComponents_A.log`

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

Using trace level logging

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

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

The LogDbg attribute should be used to enable the debug logs for the ACCLib-based agents when the ACCLIB version is 6.2.0.0 or later and the VCS version is 6.2 or later.

To localize ResLogLevel attribute for a resource

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

2 Localize the ResLogLevel attribute for the identified resource:

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

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

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

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

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

6 Note the time when the problem is reproduced.

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

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

8 Save the configuration changes.

```
# haconf -dump
```

9 Review the contents of the log file.

Use the time noted in the previous steps to diagnose the problem.

You can also contact Veritas support for more help.

To enable debug logs for all resources of type SAPComponents

◆ Enable the debug log.

```
# hatype -modify SAPComponents LogDbg DBG_5
```

To override the LogDbg attribute at resource level

- ◆ Override the LogDbg attribute at the resource level and enable the debug logs for the specific resource.

```
# hares -override SAPComponents LogDbg
# hares -modify SAPComponents LogDbg DBG_5
```

Sample Configurations

This appendix includes the following topics:

- [About sample configurations for the agents for SAPComponents](#)
- [Sample agent type definition for SAPComponents](#)
- [Sample SAP resource configuration](#)
- [Sample service group configuration for SAPComponents](#)

About sample configurations for the agents for SAPComponents

The sample configuration graphically depicts the resource types, resources, and resource dependencies within the service group. Review these dependencies carefully before configuring the agents for SAPComponents. For more information about these resource types, refer to the *Cluster Server Bundled Agents Reference Guide*.

Sample agent type definition for SAPComponents

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

Review the following examples of agent type definition files:

VCS 4.x

```
type SAPComponents (
  static str ArgList[] = { ResLogLevel, State, IState, EnvFile,
  SAPAdmin, InstProfile, ComponentType, SecondLevelMonitor,
  MonitorProgram }
  str ResLogLevel = INFO
  str EnvFile
  str SAPAdmin
  str InstProfile
  str ComponentType
  int SecondLevelMonitor = 0
  str MonitorProgram
)
```

VCS 5.0 or later

```
type SAPComponents (
  static boolean AEPTIMEOUT = 1
  static str ContainerType = Zone
  static str AgentFile = "/opt/VRTSvcs/bin/Script50Agent"
  static str AgentDirectory = "/opt/VRTSagents/ha/bin/SAPComponents"
  static str ArgList[] = { ResLogLevel, State, IState, EnvFile,
  SAPAdmin, InstProfile, ComponentType, SecondLevelMonitor,
  MonitorProgram }
  str ResLogLevel = INFO
  str EnvFile
  str SAPAdmin
  str InstProfile
  str ComponentType
  int SecondLevelMonitor = 0
  str MonitorProgram
)
```

Sample SAP resource configuration

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

Sample SAP Gateway instance

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

```
SAPComponents res1_sapgw (  
    Critical = 0  
    EnvFile = "/home/gwladm/.login"  
    SAPAdmin = gwladm  
    InstProfile = "/usr/sap/GW1/SYS/profile/GW1_W00_sapgw"  
    ComponentType = Gateway  
    SecondLevelMonitor = 1  
    MonitorProgram = "/tmp/mymonitor.sh"  
)
```

Sample SAP Web Dispatcher instance

An excerpt of the main.cf file for an SAP Web Dispatcher instance is as follows.

```
SAPComponents res1_sapwd(  
    Critical = 0  
    EnvFile = "/home/wdladm/.login"  
    SAPAdmin = wdladm  
    InstProfile = "/usr/sap/WD1/SYS/profile/WD1_W00_sapgw"  
    ComponentType = WebDispatcher  
    SecondLevelMonitor = 1  
)
```

Sample SAP Master Data Management instance

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

```
SAPComponents res1_sapmdm(  
    Critical = 0  
    EnvFile = "/home/mdmadm/.login"  
    SAPAdmin = mdmadm  
    InstProfile = "/usr/sap/MDM/SYS/profile/MDM_MDS02_rhel6sap"  
    ComponentType = MDM  
    SecondLevelMonitor = 1  
)
```

Sample service group configuration for SAPComponents

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

- The SAP Gateway, SAP Web Dispatcher, or SAP Master Data Management (MDM) component must be dependent on the database server.
- Each instance of SAPComponents (Gateway, Web Dispatcher, MDM) should have a separate virtual IP address assigned to facilitate network transparency.
- Each instance of SAPComponents (Gateway, Web Dispatcher, MDM) 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 Foundation Suite's Cluster File System (CFS).

Figure A-1 Service Group configuration for Web Dispatcher

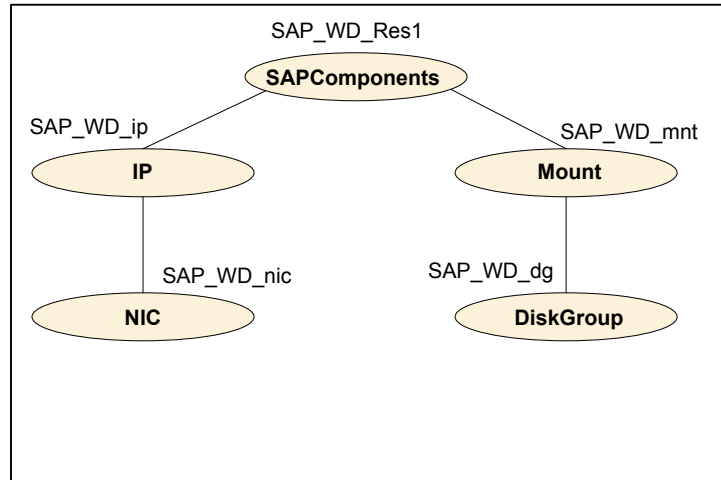


Figure A-2 Service Group configuration for SAP Gateway

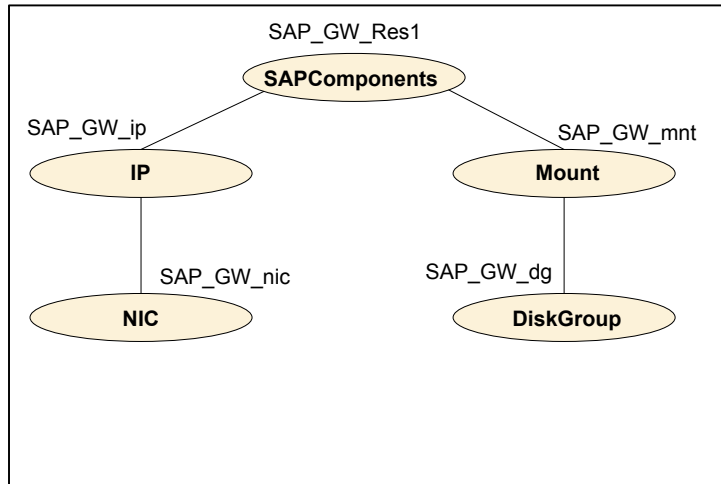


Figure A-3 Service Group configuration for SAP MDM

