Cluster Server Agent for SAP MaxDB Installation and Configuration Guide

AIX, HP-UX, Linux, Solaris

7.0

Veritas InfoScale™ Availability Agents

Last updated: 2020-10-09

Legal Notice

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

Appendix A Sample configurations

Chapter 1

Introducing the agent for SAP MaxDB

This chapter includes the following topics:

- About the Cluster Server agent for SAP MaxDB
- Supported software
- How the agent makes SAP MaxDB highly available
- SAP MaxDB agent functions
- Typical SAP MaxDB configuration in a VCS cluster
- Setting up SAP MaxDB in a VCS cluster

About the Cluster Server agent for SAP MaxDB

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

MaxDB is a relational database system that was developed for Online Transaction Processing (OLTP). The Cluster Server agent for SAP Max DB provides high availability for SAP Max DB Servers in a cluster. It is the first agent to support SAP MaxDB Server with ACC Library.

For the latest updates or software issues for this agent, see the *Cluster Server Agent Pack Release Notes*.

Supported software

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

How the agent makes SAP MaxDB highly available

The Cluster Server agent for SAP MaxDB continuously monitors the SAP MaxDB database server processes to verify that they function properly.

The agent provides the following levels of application monitoring:

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

SAP MaxDB 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:

- Verifies that the required attributes are set correctly. If the attribute values are invalid the agent responds with appropriate error message.
- Verifies whether the SAP MaxDB Server instance is not already online. If the instance is online, the online function exits immediately.
- If any SAP MaxDB Server processes remain online, the function kills these processes using the user name associated with the specific pattern `kernel` *`databaseName`*.

- Removes all the IPC resource IDs from `independentDataPath/ipc` directory if it exists.
- Starts the SAP MaxDB server with the `dbmcli` that is provided by SAP.
- Confirms that the SAP MaxDB processes starts up.
- Returns the status of SAP MaxDB agent to VCS.
- Starts the SAP MaxDB XServer and then starts the SAP MaxDB server with the `dbmcli` utility.

  or

  Starts the SAP MaxDB XServer and the `dbmcli` utility in system.slice if the UseSystemD attribute is set on the supported Linux platforms.

Offline

The offline function performs the following tasks:

- Verifies that the required attributes are set correctly. If the attribute values are invalid the agent responds with appropriate error message.
- Verifies that the SAP MaxDB Server instance is not offline. If the instance is offline, the offline function exits immediately.
- Stops the SAP MaxDB server with `dbmcli` utility provided by SAP.

  If the StopGlobalListeners attribute is set to 1, the agent stops the global listeners and XServers by using the `sdbgloballistener stop -all` command.
- Checks if any processes exist for the MaxDB Instance. If any processes exist kills them.
- Cleans up all the IPC resources for the SAP MaxDB Instance.
- Returns the exit status of SAP MaxDB agent to VCS.

Monitor

The monitor function monitors the states of the SAP MaxDB server on all nodes within the cluster. The function performs the following tasks:

- Verifies that the required attributes are set correctly. If the attribute values are invalid, the agent responds with the appropriate error messages.
- Confirms that the SAP MaxDB processes exists.
- Runs the second-level monitor attribute checks.
- Runs the `dbmcli` command to get the state of the SAP MaxDB instance if SLM is enabled.

- Validates the MonitorProgram attributes and aborts if the values are not valid.
- Runs the monitor program.
- Returns the exit status of the monitor program to the VCS High Availability Daemon (HAD).

Clean

In case of a failure or after an unsuccessful attempt to online or offline a MaxDB server instance, the clean function performs the following tasks:

- Attempts to gracefully shut down the MaxDB server instance.
  If the StopGlobalListeners attribute is set to 1, the agent attempts to gracefully shut down the global listeners and the XServers.
- If the instance does not shut down normally, the clean function kills the remaining MaxDB processes.
- Removes any existing IPC resources of the MaxDB instance.
- Runs the `dbmcli` command to clear the runtime environment for the MaxDB Instance.
- Returns the exit status to VCS.

Typical SAP MaxDB configuration in a VCS cluster

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

- VCS is installed and configured in a two-node cluster.
- The `/sapdb/DBName` directory is installed on shared storage.
- The SAP MaxDB binaries are installed locally on both nodes or on shared disks.
- The Cluster Server agent for SAP MaxDB is installed on both the nodes.

Figure 1-1 depicts a configuration where SAP MaxDB instance binaries and /sapdb/DBName are installed completely on shared disks.

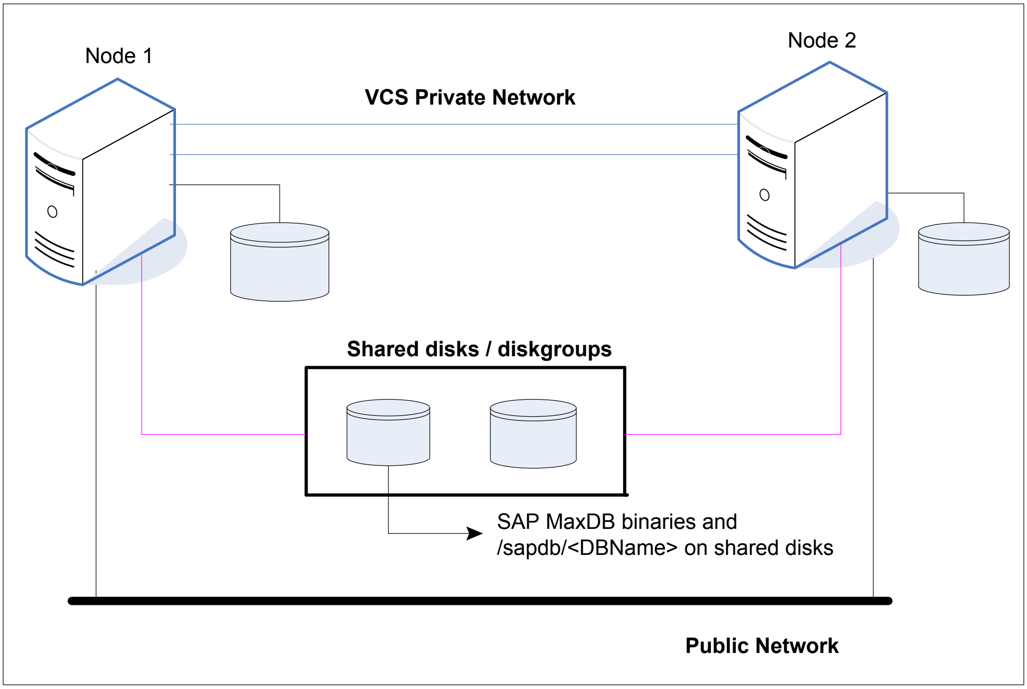
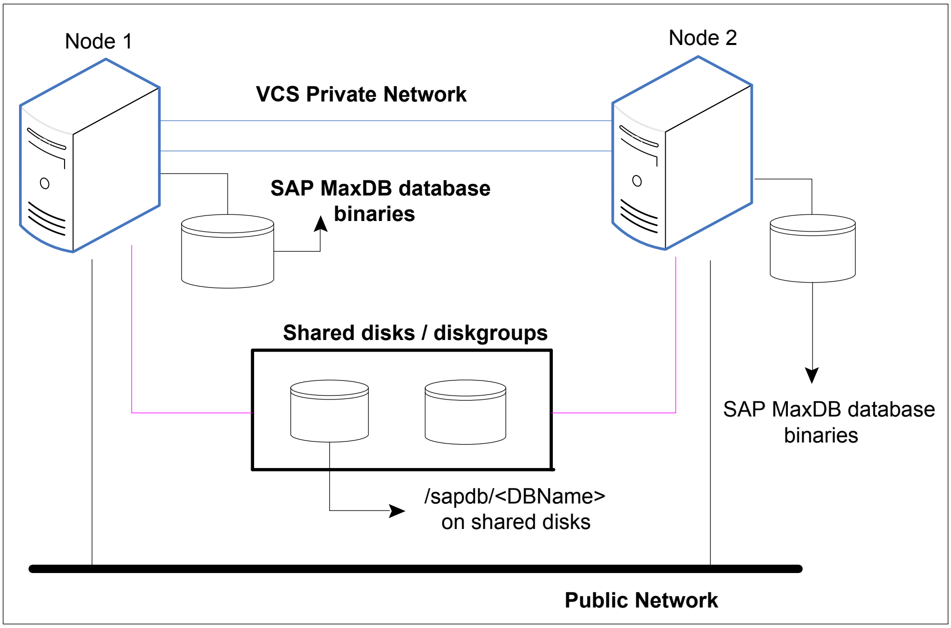


Figure 1-2 depicts a configuration where SAP MaxDB instance binaries are installed locally on each node and /sapdb/DBName is installed on shared disks.


Node 1
Node 2
VCS Private Network
SAP MaxDB database binaries
Shared disks / diskgroups
SAP MaxDB database binaries
/sapdb/<DBName> on shared disks
Public Network

Setting up SAP MaxDB in a VCS cluster

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

1. Set up a VCS cluster.
2. Install and configure SAP MaxDB for high availability.
3. Install the Cluster Server agent for SAP MaxDB.

   See “Installing the agent in a VCS environment” on page 28.
4. Configure the service groups for SAP MaxDB.

   See “About configuring service groups for SAP MaxDB” on page 44.

Installing and configuring SAP MaxDB for high availability

This chapter includes the following topics:

- About SAP MaxDB server
- Uniquely identifying SAP MaxDB server instances
- Monitoring a SAP MaxDB instance
- About configuring SAP MaxDB for high availability
- Configuring SAP MaxDB Server for cluster support

About SAP MaxDB server

SAP MaxDB is a relational database system that was developed for Online Transaction Processing (OLTP). The OLTP database instance type has been optimized for the rapid processing of individual transactions with a large number of users and large databases.

SAP MaxDB is a relational database system with which you can create, use and manage MaxDB database instances. You can use a MaxDB database as the database for an SAP system, among other uses. Prior to version 7.5, MaxDB was delivered under the name SAP DB.

SAP DB and MaxDB

In May 2003, SAP concluded a cooperation agreement with MySQL AB. The contract stipulates that, as of database version 7.5, the SAP DB database shall be delivered under the name MaxDB. MaxDB 7.5 is the result of the continuing development of the SAP DB source code. The MaxDB software 7.5 can be used as a direct upgrade for previous SAP DB versions as of 7.2.04. As per the releases for SAP applications, this also applies to SAP DB databases in SAP installations. For SAP customers, nothing changes as a result of this contract.

Overview of the Database System

SAP MaxDB is SAP's own relational database system. You can use it in SAP solutions as a less expensive alternative to databases from other vendors.

A database instance comprises a database and the additional information required for the operation of the database. A database is a set of data with a regular structure. The data in a database is comprised of the application data (data records) and the database catalog (metadata). Every database instance has a unique name. A database instance can be in different operational states, for example ONLINE, ADMIN, OFFLINE. In the ONLINE operational state, users can work with the data in the database instance, while the ADMIN operational state allows database administrators to execute administration tasks.

As a user, to query or change data in a database, you use the Structured Query Language (SQL). Using SQL, you can display, change and delete the logical units (database objects) of the database instance, such as tables. You can use schemas to group database objects logically. The database system executes SQL statements within transactions. In physical terms, a database instance comprises the data and log volumes (permanent storage), multiple caches, the files in the run directory and a collection of metadata (including configuration files, database parameters, paths of the volumes, etc.). The database system uses page chains and B* trees as logical access structures.

Accessing a MaxDB Database Instance From Remote Computers

Several different database tools help to work with database instances. To use a database instance in a database application, integrate the database instance in the database application via an interface. Database application, database tools, interfaces and database instance can be located on different computers in a network. To access a database instance located on a remote computer, the database tools and interfaces require the MaxDB XServer (communication server).

XServer

The Max DB XServer is the communication server for the database system. The XServer listens out for connection requests from clients such as database applications and database tools. The XServer is necessary if a client is establishing a connection to a database instance located on a remote computer. The JDBC interface also requires the XServer for local communication. Communication between clients and XServer can be encrypted with SSL/TLS in SAP systems. The XServer can be used on all operating systems supported by the database system. The XServer is part of the MaxDB software package.

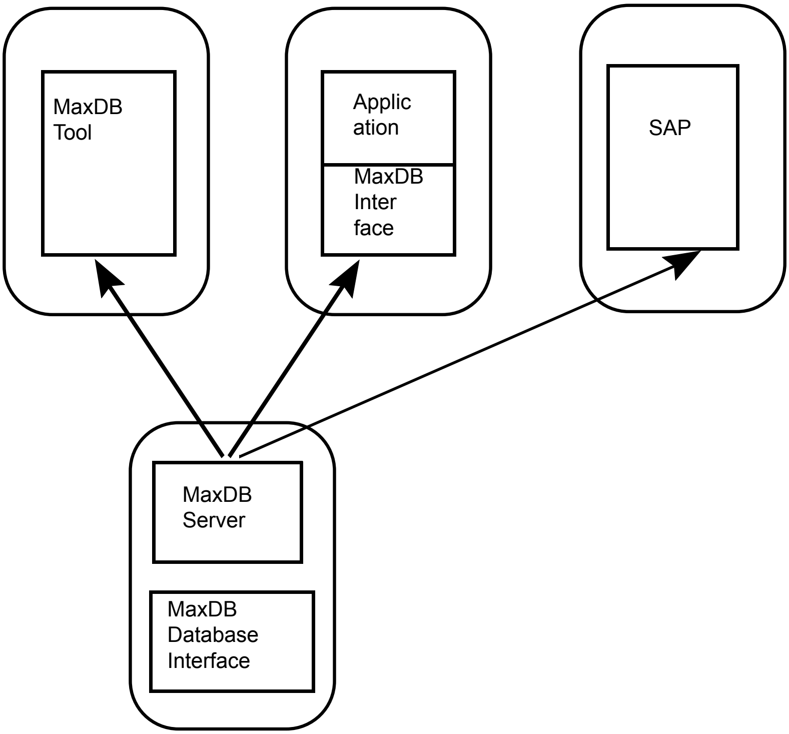
XUSER

The MaxDB XUSER database tool enables user log-in data to be stored and provides simplified log-on to database instances. You can use XUSER to store log-in data for database system administrators (SYSDBA users), database manager operators (DBM operators) and database users. XUSER is a command line tool and can be used on all operating systems supported by the database system. XUSER can be used in SAP systems. XUSER is part of the MaxDB software package.

Log-on data for logging on to database instances is defined as XUSER entry data and stored under a user key. When logging on to a database instance, you specify the user key only. XUSER entries are stored separately for each operating system user. XUSER entries are stored in the file system in UNIX/Linux. Operating system users can define up to 32 user keys for themselves. If several users are working with the same computer but log on under different user names using one database instance, individual user data can be managed separately in such cases.

Figure 2-1 illustrates how you can access a MaxDB database from remote computers.

Figure 2-1 Accessing a MaxDB database instance from remote computers



Integrating MaxDB Instances into SAP Systems

You can use MaxDB database instances in SAP systems. MaxDB can be integrated into the following SAP systems:

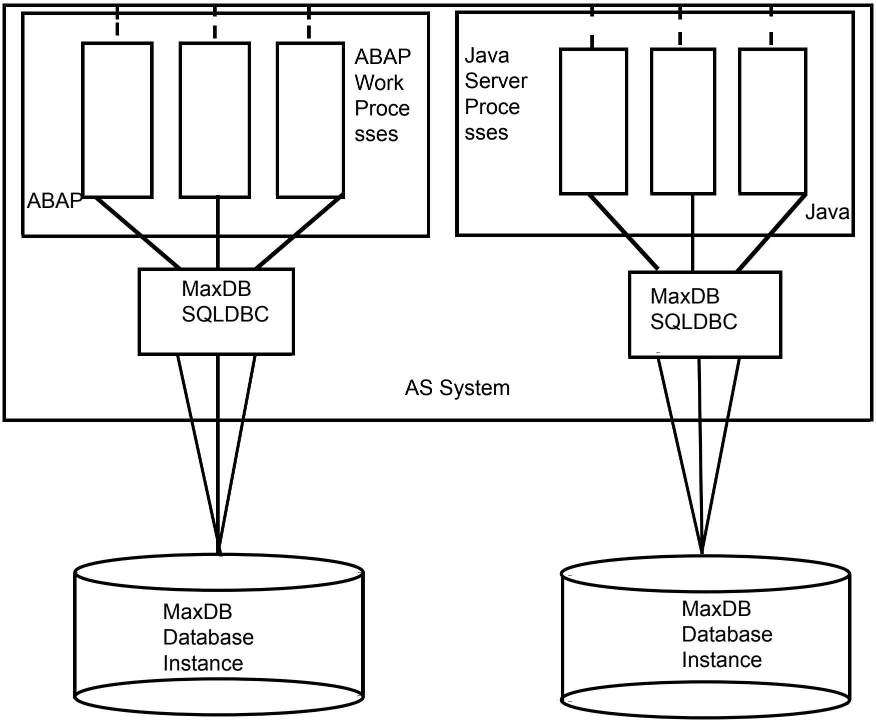
SAP NetWeaver AS	MaxDB SQLDBC
SAP NetWeaver BI	MaxDB SQL DBC
SAP NetWeaver EP	MaxDB JDBC
SAP NetWeaver MI (Clients)	MaxDB JDBC
SAP NetWeaver MDM	MaxDB JDBC
SAP NetWeaver XI	MaxDB JDBC

Integration of MaxDB into SAP NetWeaver AS

MaxDB is integrated into SAP NetWeaver AS using MaxDB SQLDBC.

Figure 2-2 shows the integration of MaxDB with SAP NetWeaver AS

Figure 2-2 Integration of MaxDB with SAP NetWeaver AS



Communication with MaxDB Server

To establish a connection to a database on a remote computer, database applications and database tools use the MaxDB X Server running on the remote computer. The X Server is concurrently available to all database instances even if several database instances are installed on one computer, only one X Server ever runs. To secure the connection between the client and the X Server, SAP customers can use SSL/TLS. Connections to database instances on the local computer use shared memory. If you want to force the database to establish the connection via the X Server nonetheless, specify localhost as the database computer when you log on to the database instance.

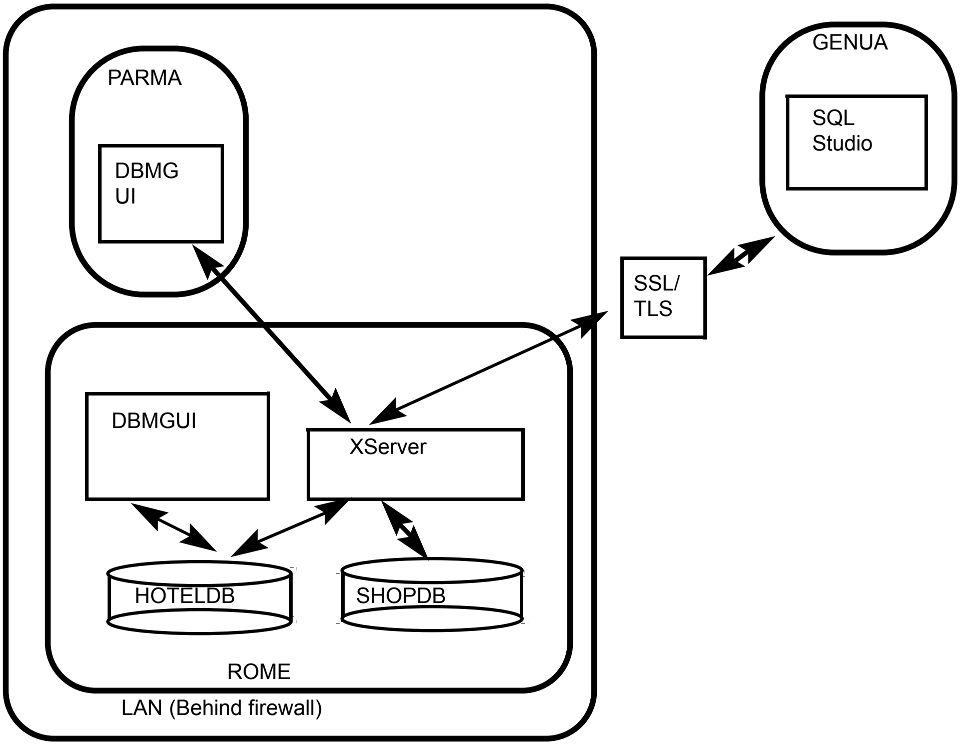
Example: Communication between Clients and Database Instance

Consider the following two database instances HOTELDB and SHOPDB that are installed on computer ROME:

- A user on computer PARMA uses the Database Manager GUI (DBMGUI) to access the database instance HOTELDB via the X Server on computer ROME.
- A second user, on computer GENUA, uses the SQL Studio to access the database instance SHOPDB via the same X Server on computer ROME.
- A third, local, user uses the Database Manager GUI to access the database instance HOTELDB without using the X Server.

Figure 2-3 shows the communication between a client and database instance

Figure 2-3 Communication between client and database Instance

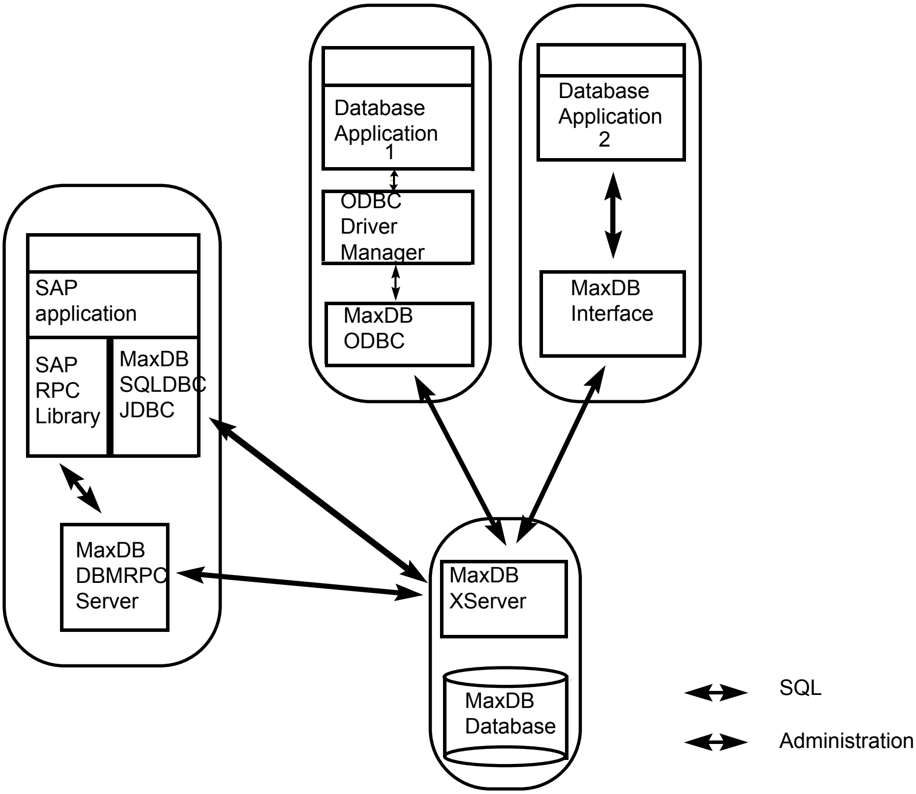


SAP MaxDB Interface

Database applications, such as SAP systems, access MaxDB database instances via interfaces. If the database application (and thus the MaxDB interface) is on a different computer than the database instance, the database system also needs the MaxDB XServer (communication server) for communication. The JDBC interface needs the MaxDB XServer even for local communication.

Figure 2-4 shows how database applications access a MaxDB Database instance via MaxDB interfaces.

Figure 2-4 Database applications accessing a MaxDB Database instance via MaxDB interfaces



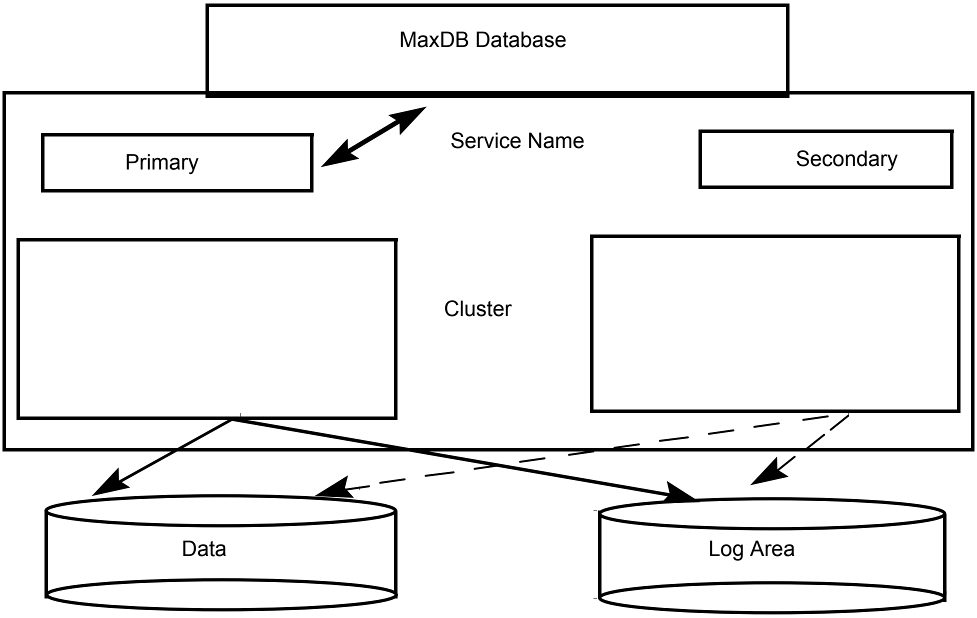
High Availability for SAP MaxDB Database

A database instance can fail due to hardware or logical errors. To restore the database instance, you have to import data and log backups. If you have a lot of data to import, restoring the database instance can take a long time. The database instance is not available for use during this time.

To make the database instance available for use more quickly (high availability), you can cluster the SAP MaxDB using VCS. VCS uses two systems for clustering SAP MaxDB. Only one system will be active for MaxDB at any point of time. When a failure occurs on the first system the file system and IP will be brought online on the second system and then MaxDB will be brought online.

Figure 2-5 shows how high availability can be achieved for SAP MaxDB database using VCS.

Figure 2-5 High availability for SAP MaxDB database



Uniquely identifying SAP MaxDB server instances

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

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

Each instance has a unique instance name.

The instance names may follow the naming conventions as follows:

- Instance name should contain only three alphanumeric uppercase characters.
- Instance name should always start with an alphabet.
- Instance name should be unique on the cluster.

Note that the DBName attribute forms a unique identifier that can identify the processes running for a particular SAP MaxDB database server instance.

For example, HOT, ERP

Monitoring a SAP MaxDB instance

The monitor operation performs process level check to ensure the proper functioning of an SAP MaxDB instance. It checks for the processes from the process table that contains "kernel DBName" in its name. It also checks the database state with "dbmcli" utility if SecondLevelMonitor attribute is set.

About configuring SAP MaxDB for high availability

The guidelines for configuring SAP MaxDB 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.

Configuring SAP MaxDB Server for cluster support

Configuring the SAP MaxDB server for cluster support involves the following phases:

Installing the MaxDB server on first node

Install the MaxDB server using SAP provided installation tool SAPInst for the required SAP application.

Complete the following steps:

To install the MaxDB server on first node

1. Start the Database Instance installation for the SAP application using SAPINST_USE_HOSTNAME=<Virtual Host Name>.

   Use the following command for starting the installation:

   ```
   sapinst SAPINST_USE_HOSTNAME=<Virtual Host Name>
   ```

   And proceed as per the instructions on the SAP installation GUI for installing Database Instance.

2. If you are installing MaxDB database for Standalone use (not with SAP applications) install the MaxDB server using SDBINST tool provided with SAP MaxDB software.

   For Standalone use install the MaxDB server with SDBINST.

   ```
   SDBINST
   ```

3. Perform all the post-installation steps described in the SAP MaxDB installation guide.

Installing the MaxDB software on second node

Install the MaxDB software on the second node with the same MaxDB name used in installing on the first node.

Complete the following steps:

To install the MaxDB server on second node

1. Create SAP MaxDB database software owner "sdb" user on the system with logon permission disabled with same UID as of first node.
2. Create SAP MaxDB database administration group "sdba" on the system with same GID as of first node.
3. Create SAP MaxDB user "<DBUser>" with same permissions as on first node and with same UID as of first node.
4. Start the MaxDB software installation from the MaxDB media with SDBINST.
5. Setup the X server logon credentials for all SAP users: <DBUser>, control, superdba as described in SAP note 39439.
6. Mount the data and log volumes on first node on to second node.
7. Start the MaxDB server.

Chapter 3

Installing, upgrading, and removing the agent for SAP MaxDB

This chapter includes the following topics:

Before you install the Cluster Server agent for SAP MaxDB

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

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

- Install and configure Cluster Server.
  For more information on installing and configuring Cluster Server, refer to the Cluster Server installation and configuration guides.

- Remove any previous version of this agent.
  To remove the agent,
  See "Uninstalling the agent in a VCS environment" on page 31.
- 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 24.

Prerequisites for enabling i18n support

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

- Install ACCLib version 5.1.4.0 or later.
  See "Installing the ACC library" on page 25.
- For VCS 5.0 and earlier releases, copy the latest ag_i18n_inc.pm module from the following location on the agent pack disc.

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

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

where *arch_dist* takes the following values:
'sol_sparc' for Solaris SPARC
'generic' for Linux

About the ACC library

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

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

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

Installing the ACC library

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

To install the ACC library

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

   You can download either the complete Agent Pack tar file or the individual ACCLib tar file from the Veritas Services and Operations Readiness Tools (SORT) site (https://sort.veritas.com/agents).
3. If you downloaded the complete Agent Pack tar file, navigate to the directory containing the package for the platform running in your environment.

AIX	*cd1*/aix/vcs/application/acc_library/*version*_library/pkgs
HP-UX	*cd1*/hpux/generic/vcs/application/acc_library/*version*_library/pkgs
Linux	*cd1*/linux/generic/vcs/application/acc_library/*version*_library/rpms
Solaris	*cd1*/solaris/*dist_arch*/vcs/application/acc_library/*version*_library/pkgs where *dist_arch* is sol_sparc.

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

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

AIX

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

HP-UX

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

Linux

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

Solaris

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

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

Installing the ACC library IPS package on Oracle Solaris 11 systems

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

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

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

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

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

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

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

4. Install the package.

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

5. Remove the publisher from the system.

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

6. Enable the publishers that were disabled earlier.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

7. Install the package.

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

8. Remove the publisher on the non-global zone.

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

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

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

10. Enable the publishers that were disabled earlier.

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

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

Installing the agent in a VCS environment

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

To install the agent in a VCS environment

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

   You can download either the complete Agent Pack tar file or an individual agent tar file.

2. Uncompress the file to a temporary location, say /tmp.

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

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

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

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

   AIX

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

   HP-UX

   ```
   # swinstall -s ‘pwd‘ VRTSsapdb
   ```

   Linux

   ```
   # rpm -ihv \
   VRTSsapdb-AgentVersion-GA_GENERIC.noarch.rpm
   ```

   Solaris

   ```
   # pkgadd -d . VRTSsapdb
   ```

   Note: See "Installing the agent IPS package on Oracle Solaris 11 systems" on page 29.

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 35.

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 `VRTSsapdb.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/VRTSsapdb.p5p Veritas
   ```

4. Install the package.

   ```
   # pkg install --accept VRTSsapdb
   ```

5. Remove the publisher from the system.

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

6. Enable the publishers that were disabled earlier.

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

Installing agent packages on Solaris brand non-global zones

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

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

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

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

2. Log on to the non-global zone as a superuser.
3. Ensure that the SMF service `svc:/application/pkg/zones-proxy-client:default` is online inside non-global zone:

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

4. Copy the `VRTSsapdb.p5p` package from the `pkgs` directory to the non-global zone (for example, at the `/tmp/install` directory).
5. Disable the publishers that are not reachable, as package install may fail, if any of the already added repositories are unreachable.

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

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

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

7. Install the package.

   ```
   # pkg install --accept VRTSsapdb
   ```

8. Remove the publisher on the non-global zone.

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

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

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

10. Enable the publishers that were disabled earlier.

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

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

Installing the agent in a Solaris 10 brand zone

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

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

Uninstalling the agent in a VCS environment

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

   ```
   # hares -list Type=SAPMaxDB
   ```

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

   ```
   # hatype -delete SAPMaxDB
   ```

   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 SAP MaxDB from each node in the cluster.

   Run the following command to uninstall the agent:

AIX	`# installp -u VRTSsapdb.rte`
HP-UX	`# swremove VRTSsapdb`
Linux	`# rpm -e VRTSsapdb`

Solaris	`# pkgrm VRTSsapdb`
	Note: To uninstall the agent IPS package on a Solaris 11 system, run the following command:
	`# pkg uninstall VRTSsapdb`

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.

```
# hagrp -freeze groupName -persistent
```

2. Stop the agent on all the cluster nodes.

```
# haagent -stop SAPMaxDB -force -sys systemName
```

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

```
# ps -ef | grep SAPMaxDB
```

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

Run the following command to uninstall the agent:

AIX	`# installp -u VRTSsapdb.rte`
HP-UX	`# swremove VRTSsapdb`
Linux	`# rpm -e VRTSsapdb`

Solaris	For Solaris 10:
	`# pkgrm VRTSsapdb`
	For Solaris 11:
	`# pkg uninstall VRTSsapdb`

5 Install the new agent on all the nodes.

See "Installing the agent in a VCS environment" on page 28.

On successful installation of the agent, the `UpdateTypes.pl` script runs automatically and adds the new attributes, if any, to the configuration.

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

```
# haagent -start SAPMaxDB -sys systemName
```

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

```
# hagrp -unfreeze groupName -persistent
```

Chapter 4

Configuring the agent for SAP MaxDB

This chapter includes the following topics:

About configuring the Cluster Server agent for SAP MaxDB

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

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

See “About sample configurations for the agents for SAP MaxDB” on page 53.

Importing the agent types files in a VCS environment

To use the agent for SAP MaxDB, 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 ■ HP-UX ■ Linux ■ Solaris	`/etc/VRTSvcs/conf/sample_SAPMaxDB/` `SAPMaxDBTypes.cf`
VCS 5.x or later	■ AIX ■ HP-UX ■ Linux	`/etc/VRTSagents/ha/conf/SAPMaxDB/` `SAPMaxDBTypes.cf`
VCS 5.0	Solaris SPARC	`/etc/VRTSagents/ha/conf/SAPMaxDB/` `SAPMaxDBTypes50.cf`
VCS 5.1 or later	Solaris SPARC	`/etc/VRTSagents/ha/conf/SAPMaxDB/` `SAPMaxDBTypes51.cf`

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

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

You can now create SAP MaxDB 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

◆ If VCS is running, execute `/etc/VRTSagents/ha/conf/SAPMaxDB/SAPMaxDBTypes.cmd`.

If VCS is not running, perform the following steps sequentially:

- Copy the agent types file from the `/etc/VRTSagents/ha/conf/`*`agentTypesFile`* directory to the `/etc/VRTSvcs/conf/config` directory. *agentTypesFile* depends on the product version and the supported operating systems:

VCS 4.x	AIX HP-UX Linux Solaris	`/etc/VRTSvcs/conf/sample_SAPMaxDB/` `SAPMaxDBTypes.cf`
VCS 5.x or later	AIX Linux	`/etc/VRTSagents/ha/conf/SAPMaxDB/` `SAPMaxDBTypes.cf`
VCS 5.0	Solaris SPARC	`/etc/VRTSagents/ha/conf/SAPMaxDB/` `SAPMaxDBTypes50.cf`
VCS 5.1 or later	Solaris SPARC	`/etc/VRTSagents/ha/conf/SAPMaxDB/` `SAPMaxDBTypes51.cf`

- Include the agent types file in the `main.cf` file.

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

- Start HAD.

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

SAP MaxDB agent attributes

SAP MaxDB service is managed within a Cluster Server (VCS) environment. VCS uses software agents to control software services within a VCS cluster. To allow VCS to monitor and control the MaxDB service effectively, the service is managed with the agent for SAP MaxDB server.

VCS deploys agents to manage all components or resources of the same type. For example, a single Mount agent will be responsible for managing all mounted file systems that are under VCS control.

The agent attributes define the specific details that will be passed from the VCS engine to the agent to uniquely identify the specific resource component that is to be managed.

Table 4-1 lists the attributes are that are passed to the SAPMaxDB agent.

Table 4-1 Required Attributes

Attribute	Description
ResLogLevel Type and dimension: string-scalar	String used to set the ResLogLevel of each instance of a resource. This attribute should not be confused with the VCS generic LogLevel type-attribute, which controls the VCS engine log level on a per agent-type basis. The ResLogLevel attribute controls the amount of ACCLib VCS agent framework based logging that is written to the VCS log file on a per resource instance basis. Valid values are: ■ ERROR : Only Error level messages are logged. ■ WARN : Above plus warning level messages are logged. ■ INFO: Above plus informational level messages are logged. This is the default log level. ■ TRACE: Above plus trace level messages are logged. This is very verbose and should only be used during diagnostic operations. Example: "TRACE" Default: "INFO"
EnvFile Type and dimension: string-scalar	Environments file for DBUser to be sourced before starting/stopping the MaxDB database server. Veritas recommends to keep the environments file on shared disk for easy maintenance. Default: " " Example: /home/sqderp/erpdbenv.csh
DBHome Type and dimension: string-scalar	Directory where the MaxDB database software is installed. This directory is needed to find the dbmcli and XServer executables. Default: " " Example: "/sapdb/programs/bin"
DBUser Type and dimension: string-scalar	Operating system user for MaxDB server. This user is responsible for starting and stopping the MaxDB database server. It must always be set to a value. In general this user takes the form sqd<DBName> or <DBName>adm. Default: " " Example: sqderp
DBName Type and dimension: string-scalar	Name of the MaxDB instance which contains a maximum of eight alphanumeric characters. Also called MaxDB SID. Only upper case letters are allowed and the first letter must be a character. This attribute is needed to uniquely identify the processes of the MaxDB database. When you use MaxDB with SAP systems the length of the DBName is three alphanumeric characters. Default: " " Example: ERP

Table 4-1 Required Attributes *(continued)*

Attribute	Description
DBHost Type and dimension: string-scalar	Virtual hostname of the MaxDB database server. Default: " " Example: saperpdb
DBMUserId Type and dimension: string-scalar	The Database Manager UserId used in the "dbmcli" to connect to the MaxDB database server. Example: control Default: "control"
DBMPassword Type and dimension: string-scalar	Password for the user <DBMUserId>. -store encrypted -agent to do decryption appropriately Use `vcsencrypt -agent` option to encrypt the password. If you are using VCS GUI to enter password no need to encrypt the password. VCS GUI will automatically encrypt password for you. Default: " " Example: jxpVmxMpkPlpMpnPo
UseSystemD Type and dimension: boolean-scalar	SystemD is a system and a service manager for Linux operating systems. It helps manage applications across Linux distributions that support the SystemD feature. ■ When this attribute is set to 1, the SAP MaxDB agent starts the SAP MaxDB processes in system.slice. The SAPMaxDB resource comes online as a service in system.slice. ■ When this attribute is set to 0, a typical online function starts the resource in user.slice.
StopGlobalListeners Type and dimension: boolean-scalar	When this attribute is set to 1, the agent stops the global listener and all the XServers via the offline and the clean entry points. Default: 0 Example: 1

Table 4-2 lists the optional attributes

Table 4-2 Optional attributes

Attribute Type and Dimension	**Definition**
MonitorProgram	Description: Full path and file name of an external, user-supplied monitor program. If specified, the monitor entry point will execute this file to perform an additional server state check. There are no restrictions for what actions the external monitor program performs to determine the state of a SAP MaxDB database. The only constraint is that the external monitor program must return one of the following integer values: ■ 0 (server is online) ■ 110 (server is online) ■ 100 (server is offline) ■ 1 (server is offline) ■ 99 or any thing other than{0,1,110,100}(server state is unknown) Veritas recommends storing the external monitor program on the shared disk directory to ensure the file is always available on the online system. Arguments are supported. Type and dimension: string-scalar Default: No default value Example1: /sapdb/data/db/wrk/ERP/mymonitor.sh Example2: /sapdb/data/db/wrk/ERP/mymonitor.sh arg1 arg2

Table 4-2 Optional attributes *(continued)*

Attribute Type and Dimension	Definition
SecondLevelMonitor	Used to enable second-level monitoring and specify how often it is run. Second-level monitoring is a deeper, more thorough state check of the configured SAP MaxDB instance. The numeric value specifies how often that the second-level monitoring routines are run. Zero (0) means never run the second-level monitoring routines. One (1) would mean to run it every monitor interval. Two (2) means to run the second-level monitoring routines every second monitor interval, and so on. Care should be taken when setting this attribute to large numbers. For example, if the MonitorInterval is set to 60 seconds, and the SecondLevelMonitor is set to 100, then the ‘dbmcli’ command for this attribute would only get executed every 100 minutes, which may not be as often as intended. In order to provide maximum flexibility, the value set is not checked for an upper limit. Thus, you could cause the SecondlevelMonitor command to run once a month, if that is what is desired. Type and dimension: integer-scalar Default: 0 Example: 1

Table 4-2 Optional attributes *(continued)*

Attribute Type and Dimension	Definition
ContainerInfo	This attribute defines the non-global zone support for VCS 5.1 on Solaris. It specifies if you can use the service group with the container. Assign the following values to the ContainerInfo attribute: Name: The name of the container. For example, sapmaxdb-zone1 Type: The type of the container. Set this to Zone. Enabled: If you want to enable the container, specify the value as 1. Else specify it as 0. For more details refer to the *Cluster Server Administrator's Guide* Type and dimension: string-association Example: {Name=sapmaxdb-zone1, Type=Zone, Enabled=1} Default: ""
ContainerName	This attribute defines the non-global zone support for VCS 5.0 on Solaris. It defines the name of the non-global zone. For more details refer to the *Cluster Server Administrator's Guide.* Type and dimension: string-association Example: {Name=sapmaxdb-zone1, Type=Zone, Enabled=1}

When you create a database instance, you define the name of the database instance (database name).

The following restrictions apply:

- The maximum length of a database name is 8 characters.
- The database name may only contain characters from the 7-Bit ASCII character set.
- The database name must not begin with an underscore or a period.
- When you create a database instance, the database system automatically changes all lowercase letters in the database name into uppercase letters.

Executing a customized monitoring program

The monitor function can execute a customized monitoring utility to perform an additional SAP MaxDB 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 MaxDB instance is online.
- The LevelTwoMonitorFreq attribute is either set to 0 or 1, and the second-level check indicates that the SAP MaxDB instance is online.

Chapter 5

Configuring the service groups for SAP MaxDB using the CLI

This chapter includes the following topics:

- About configuring service groups for SAP MaxDB
- Before configuring the service groups for SAP MaxDB
- Configuring service groups for SAP MaxDB
- Creating service groups for SAP MaxDB under Solaris non-global zones

About configuring service groups for SAP MaxDB

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

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

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

See “Configuring service groups for SAP MaxDB” on page 45.

Before configuring the service groups for SAP MaxDB

Before you configure the SAP MaxDB 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 Avaialability Installation Guide*.
- Verify that SAP MaxDB is installed and configured identically on all nodes in the cluster.
  See "About configuring SAP MaxDB for high availability " on page 21.
- Verify that the Cluster Server agent for SAP MaxDB is installed on all nodes in the cluster.

Configuring service groups for SAP MaxDB

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

1. Create a service group for SAP MaxDB.

   ```
   # hagrp -add SAP76-DBS
   ```

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

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

   For example,

   ```
   # hagrp -modify SAP76-DBS SystemList vcssx074 0 vcssx075 1
   ```

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

   ```
   # hares -add SAP76-DBS_nic NIC SAP76-DBS
   ```

   ```
   # hares -add SAP76-DBS_ip IP SAP76-DBS
   ```

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

4. Create links between the resources.

   ```
   # hares -link SAP76-DBS_ip SAP76-DBS_nic
   ```

5. Create SAPMaxDB resource for SAP MaxDB database server.

   ```
   # hares -add SAP76-DBS_db SAPMaxDB SAP76-DBS
   ```

   Based on the SAP MaxDB database server instance you are clustering, modify the resource attributes.

   For more information on agent attributes,

   See “SAP MaxDB agent attributes” on page 37.

6. Create resource dependencies for SAPMaxDB resource.

   The SAPMaxDB resource depends on the IP and Mount resources.

   ```
   # hares -link SAP76-DBS_db SAP76-DBS_ip
   ```

   ```
   # hares -link SAP76-DBS_db SAP76-DBS_mnt
   ```

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

   ```
   # hares -dep
   ```

   for example,

   ```
      Group      Parent       Child
   SAP76-DBS    SAP76-DBS_db    SAP76-DBS_mnt
   SAP76-DBS    SAP76-DBS_db    SAP76-DBS_ip
   SAP76-DBS    SAP76-DBS_ip    SAP76-DBS_nic
   SAP76-DBS    SAP76-DBS_mnt   SAP76-DBS_vol
   SAP76-DBS    SAP76-DBS_vol   SAP76-DBS_dg
   ```

Creating service groups for SAP MaxDB under Solaris non-global zones

To configure zones on each cluster node

1. Set up the non-global zone configuration.

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

   For example:

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

2. Verify the non-global zone configuration.

   ```
   hazoneverify servicegroup_name
   ```

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

4. 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`.

5. 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.
6. Configure the service groups for SAP MaxDB.

Chapter 6

Troubleshooting the agent for SAP MaxDB

This chapter includes the following topics:

- Using the correct software and operating system versions
- Meeting prerequisites
- Starting the SAP MaxDB 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 SAP MaxDB supports, see the Veritas Services and Operations Readiness Tools (SORT) site: https://sort.veritas.com/agents.

Meeting prerequisites

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

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

See "Before you install the Cluster Server agent for SAP MaxDB" on page 23.

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

You can then restart the SAP MaxDB 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 a SAP instance outside the cluster framework, is described as follows.

To restart the SAP instance outside the cluster framework

1. Log in as a superuser.
2. Use the DBUser atribute to log in to the SAP server.

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

   For certain shell versions on AIX, LOGNAME is read-only.
3. Start the SAPMaxDB server instance, using the following command:

   ```
   $ DBHome/dbmcli -d DBName -n DBHost -u DBMUserId,DBMPassword db_online
   ```

4. Ensure that the SAPMaxDB instance is running successfully by running the grep command for DBName.

   For example, for SAPMaxDB server instance ERP run the following command:

   ```
   $ ps -ef | grep ERP
   ```

   The kernel processes run on the system for MaxDB server.

   If the SAP MaxDB server instance is running outside the cluster framework, you can attempt to restart the SAP MaxDB server within the cluster framework by enabling the MaxDB cluster resource.

Reviewing error log files

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

Using SAP MaxDB log files

If a SAP MaxDB database server is facing problems, you can access the server log files to further diagnose the problem. The SAP MaxDB log files are located in the /sapdb/data/wrk/<DBName> directory.

Reviewing cluster log files

In case of problems while using the agent for SAP MaxDB, 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 SAPMaxDB agent log files located at `/var/VRTSvcs/log/SAPMaxDB_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 SAP MaxDB resource. You can set this attribute to TRACE, which enables very detailed and verbose logging.

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

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

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

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

To localize ResLogLevel attribute for a resource

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

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

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

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

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

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

8. Save the configuration changes.

   ```
   # haconf -dump
   ```

9. Review the contents of the log file.

   Use the time noted in 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 SAPMaxDB

◆ Enable the debug log.

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

Appendix A

Sample configurations

This appendix includes the following topics:

- About sample configurations for the agents for SAP MaxDB
- Sample agent type definition For MaxDB
- Sample SAP MaxDB resource configuration for VCS
- Sample service group configuration for SAP MaxDB

About sample configurations for the agents for SAP MaxDB

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

Sample agent type definition For MaxDB

For VCS 4.1

```
type SAPMaxDB (
    static str ArgList[] = { ResLogLevel, State, IState, EnvFile,
    DBHome, DBUser,     DBName, DBHost, DBMUserId, DBMPassword,
    SecondLevelMonitor,             MonitorProgram }
    str ResLogLevel = "INFO"
    str EnvFile
    str DBHome
    str DBUser
    str DBName
    str DBHost
    str DBMUserId = "control"
    str DBMPassword
    int SecondLevelMonitor
    str MonitorProgram
)
```

For VCS 5.0 or later

```
type SAPMaxDB (
    static str AgentDirectory = "/opt/VRTSagents/ha/bin/SAPMaxDB"
    static str AgentFile = "/opt/VRTSvcs/bin/Script50Agent"

    static str ArgList[] = { ResLogLevel, State, IState,
      EnvFile, DBHome, DBUser, DBName, DBHost, DBMUserId,
      DBMPassword, SecondLevelMonitor, MonitorProgram,
      UseSystemD, StopGlobalListeners }
    str ResLogLevel = "INFO"
    str EnvFile
    str DBHome
    str DBUser
    str DBName
    str DBHost
    str DBMUserId = "control"
    str DBMPassword
    int SecondLevelMonitor
    str MonitorProgram
    boolean UseSystemD = 0
    boolean StopGlobalListeners = 0
)
```

Sample SAP MaxDB resource configuration for VCS

A sample resource configuration for SAP MaxDB is as follows:

```
SAPMaxDB Resource_MaxDB (
    ResLogLevel     = INFO
    EnvFile      = /home/sqderp/erpdbenv.csh
    DBHome     = /sapdb/programs/bin
    DBUser       = sqderp
    DBName     = ERP
    DBHost       = saperpdb
```

```
        DBMUserId      = control
        DBMPassword      = xdfaasaskfaasei
        SecondLevelMonitor    = 0
        MonitorProgram     = /sapdb/data/wrk/ERP/mymonitor.sh
        UseSystemD = 1
        StopGlobalListeners = 1
)
```

Sample service group configuration for SAP MaxDB

Create SAP MaxDB resource using VCS GUI.

Figure A-1 shows a sample configuration of VCS resource for the SAP MaxDB Server.

Figure A-1 SAP MaxDB service group configuration

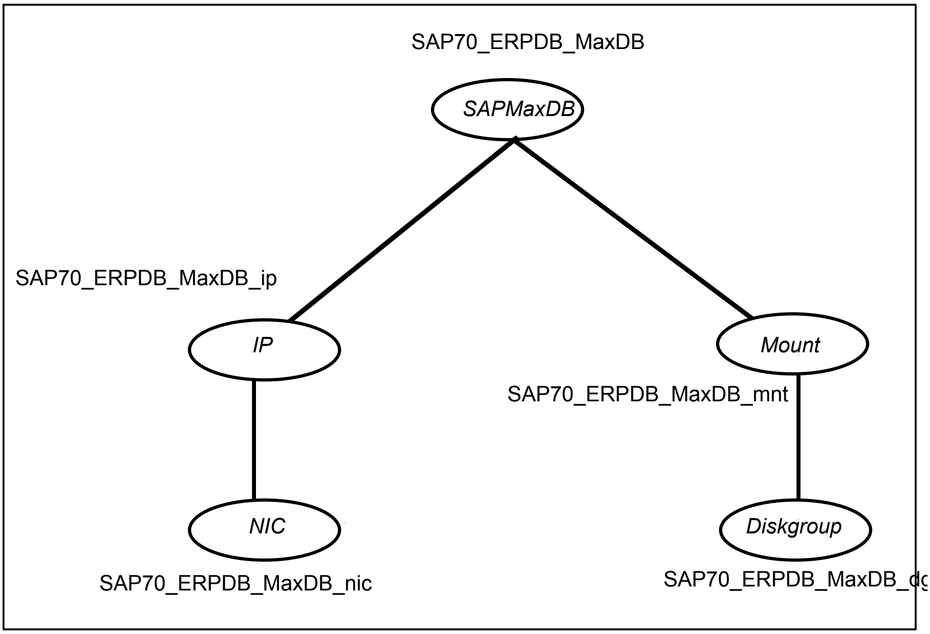


Table A-1 depicts a typical resource configuration for SAPMaxDB.

Table A-1 SAPMaxDB resource configuration

Attribute	Value
ResLogLevel	INFO
EnvFile	/home/sqderp/saperpdb.csh
DBHome	/sapdb/programs/bin
DBUser	sqderp
DBName	ERP
DBHost	saperpdb
DBMUserId	control
DBMPassword	iwoUlwLojOkoLomOn
SecondLevelMonitor	0
MonitorProgram	
UseSystemD	0
StopGlobalListeners	0