Symantec™ ApplicationHA Agent for Oracle Configuration Guide

Linux on KVM

6.1

October 2013

Symantec

Symantec™ ApplicationHA Agent for Oracle Configuration Guide

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

Agent version: 6.1.000.000

Document version: 6.1.000.000 Rev 0

Legal Notice

Copyright © 2013 Symantec Corporation. All rights reserved.

Symantec, the Symantec Logo, the Checkmark Logo, Veritas, Veritas Storage Foundation, CommandCentral, NetBackup, Enterprise Vault, and LiveUpdate are trademarks or registered trademarks of Symantec Corporation or its affiliates in the U.S. and other countries. Other names may be trademarks of their respective owners.

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

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. SYMANTEC CORPORATION SHALL NOT BE LIABLE FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES IN CONNECTION WITH THE FURNISHING, PERFORMANCE, OR USE OF THIS DOCUMENTATION. THE INFORMATION CONTAINED IN THIS DOCUMENTATION IS SUBJECT TO CHANGE WITHOUT NOTICE.

The Licensed Software and Documentation are deemed to be commercial computer software 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, "Rights in Commercial Computer Software or Commercial Computer Software Documentation", as applicable, and any successor regulations, whether delivered by Symantec 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.

Symantec Corporation
350 Ellis Street
Mountain View, CA 94043

http://www.symantec.com

Technical Support

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

Symantec's support offerings include the following:

- A range of support options that give you the flexibility to select the right amount of service for any size organization
- Telephone and/or Web-based support that provides rapid response and up-to-the-minute information
- Upgrade assurance that delivers software upgrades
- Global support purchased on a regional business hours or 24 hours a day, 7 days a week basis
- Premium service offerings that include Account Management Services

For information about Symantec's support offerings, you can visit our website at the following URL:

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

All support services will be delivered in accordance with your support agreement and the then-current enterprise technical support policy.

Contacting Technical Support

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

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

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

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

- Product release level
- Hardware information

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

Licensing and registration

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

www.symantec.com/business/support/

Customer service

Customer service information is available at the following URL:

www.symantec.com/business/support/

Customer Service is available to assist with non-technical questions, such as the following types of issues:

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

Support agreement resources

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

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

Documentation

Product guides are available on the media in PDF format. Make sure that you are using the current version of the documentation. The document version appears on page 2 of each guide. The latest product documentation is available on the Symantec website.

https://sort.symantec.com/documents

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

doc_feedback@symantec.com

For information regarding the latest HOWTO articles, documentation updates, or to ask a question regarding product documentation, visit the Storage and Clustering Documentation forum on Symantec Connect.

https://www-secure.symantec.com/connect/storage-management/forums/storage-and-clustering-documentation

About Symantec Connect

Symantec Connect is the peer-to-peer technical community site for Symantec's enterprise customers. Participants can connect and share information with other product users, including creating forum posts, articles, videos, downloads, blogs and suggesting ideas, as well as interact with Symantec product teams and Technical Support. Content is rated by the community, and members receive reward points for their contributions.

http://www.symantec.com/connect/storage-management

Contents

Technical Support 4

Chapter 1 Introducing the Symantec ApplicationHA agent for Oracle 9

About the Symantec ApplicationHA agent for Oracle 9
Changes introduced in the 6.1 release 10
About installing and removing the ApplicationHA agent for Oracle 10
Supported software 10
Supported application versions 11
Supported virtualization environments 11
Supported operating systems on virtual machines 11
How the agent makes Oracle highly available 12
About Symantec ApplicationHA agent for Oracle functions 13
Oracle agent functions 13
How the Oracle agent supports health check monitoring 18
Netlsnr agent functions 21
Typical Oracle configuration in an ApplicationHA virtual machine 21
About setting up Oracle in an ApplicationHA virtual machine 22

Chapter 2 Installing and configuring Oracle 23

About installing Oracle in an ApplicationHA environment 23
Before you install Oracle in an ApplicationHA environment 24
About ApplicationHA requirements for installing Oracle 24

Chapter 3 Configuring application monitoring with Symantec ApplicationHA 27

About configuring application monitoring with ApplicationHA 27
Before configuring application monitoring for Oracle 28
Accessing the Symantec High Availability view 28
Configuring application monitoring for Oracle 29

Chapter 4 Troubleshooting Symantec ApplicationHA agent for Oracle 34

About troubleshooting ApplicationHA agent for Oracle 34
Error messages common to the Oracle and Netlsnr agents 35
Error messages specific to the Oracle agent 36
Error messages specific to the Netlsnr agent 40
Verifying the Oracle health check binaries and intentional offline for an instance of Oracle 41
Verifying the intentional offline behavior of the VCS Oracle agent 43

Appendix A Resource type definitions 45

About the resource type and attribute definitions 45
Resource type definition for the Oracle agent 45
Attribute definition for the Oracle agent 46
Resource type definition for the Netlsnr agent 52
Attribute definition for the Netlsnr agent 52

Appendix B Detail monitoring 56

Setting the PATH variable 56
Setting up detail monitoring for ApplicationHA agent for Oracle 56
Setting up detail monitoring for Oracle 57
Setting up detail monitoring for ApplicationHA agent for Netlsnr 60

Appendix C Using the SPFILE in an ApplicationHA system for Oracle 61

Using the SPFILE in an ApplicationHA virtual machine 61

Appendix D Best practices 63

Best practices for multiple Oracle instance configurations in a ApplicationHA environment 63

Chapter 1

Introducing the Symantec ApplicationHA agent for Oracle

This chapter includes the following topics:

- About the Symantec ApplicationHA agent for Oracle
- Changes introduced in the 6.1 release
- About installing and removing the ApplicationHA agent for Oracle
- Supported software
- How the agent makes Oracle highly available
- About Symantec ApplicationHA agent for Oracle functions
- Typical Oracle configuration in an ApplicationHA virtual machine
- About setting up Oracle in an ApplicationHA virtual machine

About the Symantec ApplicationHA agent for Oracle

The ApplicationHA agent for Oracle provides controlling and monitoring for Oracle database instances.

Symantec agents do the following:

- Monitor specific resources within an enterprise application.
- Determine the status of these resources.
- Start or stop the resources according to external events.

The agents include resource type declarations and agent executables. The agent for Oracle monitors the Oracle and listener processes, brings them online, and takes them offline.

The ApplicationHA package contains the following agents that work together to make Oracle highly available:

- The Oracle agent monitors the Oracle database processes.
- The Netlsnr agent monitors the listener process.

Changes introduced in the 6.1 release

Symantec ships the `build_oraapi.sh` script along with the VCS agent for Oracle in the Oracle agent's `/opt/VRTSagents/ha/bin/Oracle` directory. The `build_oraapi.sh` script can be used to link the libraries that are required for Oracle health check monitoring.

Note: In this release, the health check monitoring option is not supported for the ASMInst agent.

About installing and removing the ApplicationHA agent for Oracle

When you install or uninstall Symantec ApplicationHA, the ApplicationHA agent for Oracle is automatically installed or removed. For more information, see the *Symantec ApplicationHA Installation and Upgrade Guide*.

When you run the installer or uninstall program that accompanies the quarterly agent pack release of high availability agents from Symantec, the latest version of the ApplicationHA agent for Oracle is automatically installed or removed. For more information, see the *Symantec ApplicationHA Agent Pack Installation Guide*.

Supported software

The Symantec ApplicationHA agent for Oracle supports the following software versions:

- Symantec ApplicationHA agent for Oracle can be installed and run inside virtual machines that have Symantec ApplicationHA 6.1 installed.
- The following versions of the Veritas Operations Manager components are supported:

- Veritas Operations Manager Management Server 6.0 or later
- Veritas Operations Manager managed host for Linux: 6.0 or later

Supported application versions

Table 1-1 lists the Oracle versions that Symantec ApplicationHA 6.1 currently supports on virtual machine.

Table 1-1 Supported application versions

Application	Version
Oracle	10gR2, 11gR1, 11gR2, and 12c

Supported virtualization environments

Symantec ApplicationHA can be installed and run inside virtual machines in a KVM virtualization environment, running Red Hat Enterprise Linux (RHEL) 6, Update 3 and 4 in the physical host.

Supported operating systems on virtual machines

Table 1-2 shows the supported operating systems for Symantec ApplicationHA 6.1.

Table 1-2 Supported guest operating systems

Operating systems	Levels	Kernel version
Red Hat Enterprise Linux 5	Updates 5, 6, 7, 8, 9	2.6.18-194.el5 2.6.18-238.el5 2.6.18-274.el5 2.6.18-308.el5 2.6.18-348.el5
Red Hat Enterprise Linux 6	Updates 3 and 4	2.6.32-279.el6 2.6.32-358.el6

Note: 64-bit operating systems are only supported.

If your system is running a lower level of Red Hat Enterprise Linux, than indicated in Table 1-2, you must upgrade it before attempting to install Symantec

ApplicationHA. Consult the Red Hat documentation for more information on upgrading or reinstalling your operating system.

Symantec supports only Red Hat distributed kernel binaries.

Symantec products operate on subsequent kernel and patch releases provided the operating systems maintain kernel ABI (application binary interface) compatibility.

How the agent makes Oracle highly available

The Symantec ApplicationHA agent for Oracle continuously monitors the Oracle database and listener processes to verify 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 Oracle and listener processes are present in the process table. Process check cannot detect whether processes are in a hung state or a stopped state.
  The Oracle agent provides functionality to detect whether the Oracle resource was intentionally taken offline. The agent detects graceful shutdown for Oracle 10g and later. When an administrator brings down Oracle gracefully, the agent does not trigger a resource fault even though Oracle is down. The value of the type-level attribute IntentionalOffline and MonitorOption must be set to 1 to enable Oracle agent's intentional offline functionality.
  For example, with the intentional offline functionality, the agent faults the Oracle resource if there is an abnormal termination of the instance. The agent reports the Oracle resource as offline if you gracefully bring down Oracle using commands like `shutdown`, `shutdown immediate`, `shutdown abort`, or `shutdown transactional`.
- Secondary or Detail monitoring
  In this mode, the agent runs a perl script that executes commands against the database and listener to verify their status.

The agent detects application failure if the monitoring routine reports an improper function of the Oracle or listener processes. When this application failure occurs, the agents attempt to start the application services. After a configurable number of attempts, if the application services do not start, the agents consider this as an application failure and report the status to VCS. Depending on the configuration, VCS can then restart the virtual machine. After the machine restarts, the agents starts the application services and brings the configured resources online. The agent thus ensures high availability for the Oracle services and the database.

Note: You can use the ApplicationHA wizards to configure only primary or basic monitoring. To configure secondary or detailed monitoring, use CLI/Veritas Operations Manager (VOM).

About Symantec ApplicationHA agent for Oracle functions

The functions an agent performs are called entry points. Review the functions for the following agents that are part of the Symantec ApplicationHA agent suite for Oracle:

- Oracle agent functions
  See "Oracle agent functions" on page 13.
- Netlsnr agent functions
  See "Netlsnr agent functions" on page 21.

Oracle agent functions

The Oracle agent monitors the database processes.

Table 1-3 lists the Oracle agent functions.

Table 1-3 Oracle agent functions

Agent operation	Description
Online	Starts the Oracle database by using the following `sqlplus` command: `startup force pfile=$PFile` The default Startup option is STARTUP_FORCE. You can also configure the agent to start the database using different Startup options for Oracle. See "Startup and shutdown options for the Oracle agent" on page 14.

Table 1-3 Oracle agent functions *(continued)*

Agent operation	Description
Offline	Stops the Oracle database with the specified options by using the following `sqlplus` command: `shutdown immediate` The default Shutdown option is IMMEDIATE. You can also configure the agent to stop the database using different Shutdown options for Oracle. See "Startup and shutdown options for the Oracle agent" on page 14.
Monitor	Verifies the status of the Oracle processes. The Oracle agent provides two levels of monitoring: basic and detail. See "Monitor options for the Oracle agent" on page 16.
Clean	Forcibly stops the Oracle database by using the following `sqlplus` command: `shutdown abort` If the process does not respond to the `shutdown` command, then the agent does the following: ■ Scans the process table for the processes that are associated with the configured instance ■ Kills the processes that are associated with the configured instance

Startup and shutdown options for the Oracle agent

You can specify Startup and Shutdown options for the Oracle instances that are configured.

Table 1-4 lists the startup options that the agent supports.

Table 1-4 Startup options

Option	Description
STARTUP_FORCE (Default)	Runs the command `startup force pfile='location_of_pfile'` if the pfile is configured. If the pfile is not configured, the agent runs `startup force`. It picks up the default parameter files from their default locations.

Table 1-4 Startup options *(continued)*

Option	Description
STARTUP	Runs the command `startup pfile='location_of_pfile'` if the pfile is configured. If the pfile is not configured, the agent picks up the default parameter files from their default locations and runs `startup`.
RESTRICTED	Starts the database in the RESTRICTED mode.
RECOVERDB	Performs a database recovery on instance startup.

CUSTOM	Uses a predefined SQL script (`start_custom_$SID.sql`) and runs custom startup options. The script must be in the /opt/VRTSagents/ha/bin/Oracle directory and must have access to the Oracle Owner OS user. If the file is not present, the agent logs an error message.
	With a custom script, the agent takes the following action:
	`sqlplus /nolog <<!`
	`connect / as sysdba;`
	`@start_custom_$SID.sql`
	`exit;`
	`!`

SRVCTLSTART	Uses the `srvctl` utility to start an instance of the database. For RAC clusters, you must manually set the default startup option as SRVCTLSTART.
SRVCTLSTART_RO	Uses the `srvctl` utility to start an instance of the database with the `-o 'read only'` option.

Table 1-5 lists the shutdown options that the agent supports.

Table 1-5 Shutdown options

Option	Description
IMMEDIATE (Default)	Shuts down the Oracle instance by running `shutdown immediate`.
TRANSACTIONAL	Runs the `shutdown transactional` command. This option is valid only for the database versions that support this option.

Table 1-5 Shutdown options *(continued)*

Option	Description
CUSTOM	Uses a predefined SQL script (`shut_custom_$SID.sql`) and runs custom shutdown options. The script must be in the /opt/VRTSagents/ha/bin/Oracle directory and must have access to the Oracle Owner OS user. If the file is not present, the agent shuts the agent down with the default option.
SRVCTLSTOP	Uses the `srvctl` utility to stop an instance of the database. For RAC clusters, you must manually set the default option as SRVCTLSTOP.
SRVCTLSTOP_TRANSACT	Uses the `srvctl` utility to stop an instance of the database with the `-o transactional` option.
SRVCTLSTOP_ABORT	Uses the `srvctl` utility to stop an instance of the database with the `-o abort` option.
SRVCTLSTOP_IMMEDIATE	Uses the `srvctl` utility to stop an instance of the database with the `-o immediate` option.

Monitor options for the Oracle agent

The Oracle agent provides two levels of monitoring: basic and detail. By default, the agent does a basic monitoring.

The basic monitoring mode has the following options:

- Process check
- Health check

The MonitorOption attribute of the Oracle resource determines whether the agent must perform basic monitoring in Process check or Health check mode.

Table 1-6 describes the basic monitoring options.

Table 1-6 Basic monitoring options

Option	Description
0 (Default)	Process check The agent scans the process table for the ora_dbw, ora_smon, ora_pmon, and ora_lgwr processes to verify that Oracle is running.

Table 1-6 Basic monitoring options *(continued)*

Option	Description
1	Health check (supported on Oracle 10g and later) The agent uses the Oracle health check APIs to monitor the System Global Area (SGA) and retrieve the information about the instance. If you want to use the Oracle agent's intentional offline functionality, you must enable health check monitoring.

Review the following considerations if you want to configure basic monitoring:

- When the administrator brings an Oracle database online on the ApplicationHA virtual machine, an Oracle instance begins, but the database is not mounted. In such circumstances, this failure is detected only by the health check monitoring option of basic monitoring or detail monitoring. Detail monitoring updates the database table after detecting a failure whereas health check monitoring does not.
  If process check monitoring of basic monitoring is set, then such a conflict would go undetected. To set health check monitoring option, use the command line interface (CLI) or VOM.

In the detail monitoring mode, the agent performs a transaction on a test table in the database to ensure that Oracle functions properly. The agent uses this test table for internal purposes. Symantec recommends that you do not perform any other transaction on the test table. The DetailMonitor attribute of the Oracle resource determines whether the agent must perform detail monitoring.

Note: Use the ApplicationHA wizards to configure basic monitoring for Oracle. To enable detail monitoring, use CLI or VOM.

How the agent handles Oracle error codes during detail monitoring

The Symantec ApplicationHA agent for Oracle handles Oracle errors during detail monitoring. The agent classifies Oracle errors according to their severity and associates predefined actions with each error code.

The agent includes a reference file oraerror.dat, which lists Oracle errors and the action to be taken when the error is encountered.

The file stores information in the following format:

```
Oracle_error_string:action_to_be_taken
```

For example:

```
01035:WARN
01034:FAILOVER
```

Note: When the oraerror.dat file contains two or more errors, then the errors take an ascending precedence and the ApplicationHA agent for Oracle handles only the first error.

When the oraerror.dat file contains two or more errors, the order in which the ApplicationHA agent for Oracle handles the errors is: FAILOVER/NOFAILOVER, UNKNOWN, and IGNORE/WARN.

Table 1-7 lists the predefined actions that the agent takes when an Oracle error is encountered.

Table 1-7 Predefined agent actions for Oracle errors

Action	Description
IGNORE	Ignores the error. When the Symantec agent for Oracle encounters an error, the agent matches the error code in the oraerror.dat file. If the error does not have a matching error code in the file, then the agent ignores the error.
UNKNOWN	Marks the resource state as UNKNOWN and sends a notification if the Notifier resource is configured. This action is typically associated with configuration errors or program interface errors.
WARN	Marks the resource state as ONLINE and sends a notification if the Notifier resource is configured. This action is typically associated with errors due to exceeded quota limits, session limits/restricted sessions so on.

How the Oracle agent supports health check monitoring

The Symantec Cluster Server agent for Oracle uses the Oracle health check APIs to determine whether the Oracle instance on a node was shut down gracefully or was it aborted. When an oracle instance is shut down gracefully outside of VCS control the agent acknowledges the operation as intentional offline.

The Oracle health check APIs are supported on all UNIX platforms, Oracle Database 10g Release and later.

You can use the `build_oraapi.sh` script to link the libraries that are required for Oracle health check monitoring. Symantec ships the `build_oraapi.sh` script along with the VCS agent for Oracle in the Oracle agent's `/opt/VRTSagents/ha/bin/Oracle` directory.

Generating the executable required for Oracle health check monitoring

Review the considerations before using the `build_oraapi.sh` script.

Prerequisites to link Oracle libraries using the `build_oraapi.sh` script

Ensure that the following prerequisites are met before building the Oracle health check libraries:

- Set the ORACLE_HOME environment variable.

  ```
  # export ORACLE_HOME=/u01/oraHome
  ```

  where */u01/oraHome* is the directory path where the *ORACLE_HOME* environment variable is set.

- If the *ORACLE_HOME* variable is not set, use the directory path to the *ORACLE_HOME* variable as a command line option along with the Oracle version when you run the script.

  ```
  # build_oraapi.sh -version <version> [-oracle_home
  <path_to_oracle_home>]
  ```

  Where *version* is the Oracle database version and *path_to_oracle_home* is the path to access the ORACLE_HOME variable.

- Ensure that the *ORACLE_HOME* variable is accessible before the executing the `build_oraapi.sh` script.

Considerations to run the build_oraapi.sh script

- Run the `build_oraapi.sh` script on each node of the cluster where you want to use Oracle health monitoring.
- You need to run the `build_oraapi.sh` script to relink the Oracle libraries whenever the Oracle database version is upgraded on a node.
- The Oracle health check binaries generated in the Oracle agent directory correspond to the `ORACLE_HOME` directory and the Oracle database version that are specified while executing the `build_oraapi.sh` script. ORACLE_HOME can be set as an environment variable or specified as a command line option.

The Oracle database version can be specified as a command line option with the `build_oraapi.sh` script.

- Symantec ships the build_oraapi.sh script in VCS 6.1 release. Symantec will not ship the pre-built Oracle health check binaries (oraapi_32, oraapi_3211g, oraapi_64, and oraapi_6411g).
- To enable health check support in Oracle agent, you must run the `build_oraapi.sh` script to build the Oracle health check binaries based on Oracle version.

Building the Oracle agent executable using the build_oraapi.sh script

To build the Oracle agent executable using the build_oraapi.sh script:

1. Log in as root user on the node.
2. Set the Oracle home directory path.

   ```
   # export ORACLE_HOME=/u01/oraHome
   ```

   where, */u01/oraHome* is the Oracle home directory that stores files.
3. Access the Oracle agent directory.

   ```
   # cd /opt/VRTSagents/ha/bin/Oracle
   ```
4. Run the following command from the Oracle agent directory.

   ```
   # ./build_oraapi.sh -version version
   ```

   where *version* is the Oracle database version on the node.
5. Alternatively, if you did not set the *ORACLE_HOME* variable, you can set the *oracle_home* parameter as a command line option.

   ```
   #./build_oraapi.sh -version version -oracle_home /u01/oraHome
   ```

   where *version* is the Oracle database version on the node and */u01/oraHome* is the Oracle home directory path, which is set as the *ORACLE_HOME* variable value.

The script generates binaries under agent directory `/opt/VRTSagents/ha/bin/Oracle`. Oracle health check binaries generated on a node is based on the Oracle major version (10, 11, or 12) and the system architecture (32 bit or 64 bit) available on that node.

For example, `oraapi_6411g` is a binary created with Oracle version 11 on a 64-bit system.

See "Verifying the Oracle health check binaries and intentional offline for an instance of Oracle " on page 41.

Netlsnr agent functions

The listener is a server process that listens to incoming client connection requests and manages traffic to the database. The Netlsnr agent brings the listener services online, monitors their status, and takes them offline.

Table 1-8 lists the Netlsnr agent functions.

Table 1-8 Netlsnr agent functions

Agent operation	Description
Online	Starts the listener process by using the following command: `lsnrctl start $LISTENER`
Offline	Stops the listener process by using the following command: `lsnrctl stop $LISTENER` If the listener is configured with a password, the agent uses the password to stop the listener.
Monitor	Verifies the status of the listener process. The Netlsnr agent provides two levels of monitoring, basic and detail: ■ In the basic monitoring mode, the agent scans the process table for the `tnslsnr` process to verify that the listener process is running. ■ In the detail monitoring mode, the agent uses the `lsnrctl status $LISTENER` command to verify the status of the Listener process. (Default)
Clean	Scans the process table for `tnslsnr $Listener` and kills it.

Typical Oracle configuration in an ApplicationHA virtual machine

A typical Oracle configuration in an ApplicationHA environment has the following characteristics:

- Oracle binaries can be installed locally or on shared disks. Data can reside on data disks. Ensure that the data disks and oracle installation binary disks are automatically mounted after the virtual machine starts/reboots. To automatically mount the required disks, add appropriate entries in `/etc/fstab`

About setting up Oracle in an ApplicationHA virtual machine

Tasks involved in setting up Oracle in an ApplicationHA environment include:

- Setting up an ApplicationHA virtual machine
  Refer to *Symantec ApplicationHA Installation Guide* for more information on installing and configuring ApplicationHA.
- Installing and configuring Oracle
  See "About installing Oracle in an ApplicationHA environment" on page 23.

Chapter 2

Installing and configuring Oracle

This chapter includes the following topics:

- About installing Oracle in an ApplicationHA environment
- Before you install Oracle in an ApplicationHA environment
- About ApplicationHA requirements for installing Oracle

About installing Oracle in an ApplicationHA environment

The strategy for installing Oracle into an ApplicationHA virtual machine is aimed at ensuring that installations on all virtual machines are uniform.

See the Oracle documentation on Linux.

You can install Oracle in the following ways in an ApplicationHA environment:

$ORACLE_HOME on the shared disk	Oracle binaries and Oracle data are installed on shared disks.
$ORACLE_HOME on the local disk	Oracle binaries are installed locally on each node and Oracle data is installed on shared disks.

Note that Oracle data includes the datafiles, control files, redo log files, and archive log files.

The user oracle and the group dba must be local and not Network Information Service (NIS and NIS+) users.

Before you install Oracle in an ApplicationHA environment

Make sure you meet the following prerequisites:

- Verify that all virtual machines have adequate resources to run Oracle and ApplicationHA.
- Verify that the network supports the TCP/IP protocol.
- Make sure that you meet the ApplicationHA requirements to install Oracle. See "About ApplicationHA requirements for installing Oracle" on page 24.

About ApplicationHA requirements for installing Oracle

Make sure you meet the following requirements to install Oracle in an ApplicationHA virtual machine.

Kernel parameter configuration	Each node on which you want to install Oracle must meet the following Oracle configuration requirements: ■ Disk partitions ■ Shared memory ■ Swap size ■ Semaphores ■ File handles See Oracle documentation for the corresponding operating system for specific requirement details.
Linux package group	Symantec recommends you to select the Software Development package option when you install Linux. This option is required for relinking Oracle at install time and to take advantage of some Advanced Server features.
JDK requirements for Linux	Make sure that Sun's JDK 1.3.1_02 or Blackdown's JDK 1.1.8 is installed on the system.

Location of the $ORACLE_HOME	Depending on your environment, you can place the Oracle home directory ($ORACLE_HOME) in one of the following ways: ■ Locally on each server's disk ■ On the shared storage. Review the advantages of each approach to make a decision.
Configurations with multiple Oracle instances (SIDs)	You can have multiple Oracle instances on a virtual machine. In such cases, the parameter file for each instance must be accessible on all virtual machines. **Note:** If you installed multiple versions of Oracle on the same system, make sure that the SIDs are unique.

Location of Oracle database tablespaces	If you plan to create the tablespaces using regular (EXT2/EXT3 or VxFS) files, the file systems that contain these files must be located on shared disks. Create the same file system mount points on each node.
	If you use raw devices on shared disks for Oracle tablespaces, you must meet the following requirements:
	■ The ownership must be Oracle DBA user.
	■ The permissions or access mode must be 660 on the raw devices that store the Oracle data.
	For example, if you use Veritas Volume Manager, type:
	`# vxedit -g diskgroup_name set group=dba \`
	`user=oracle mode=660 volume_name`
	Note: The user `oracle` and the group `dba` must be local and not Network Information Service (NIS and NIS+) users

Listener authentication in ApplicationHA environment	The Netlsnr agent supports OS authentication as well as password authentication for the listener process. If you use Oracle 10g or later, Symantec recommends you to configure OS authentication. If you want to configure a listener password, make sure that you configure the password correctly. A misconfigured password can cause the listener to fault. Refer to the Oracle documentation for details on configuring the listener authentication.
Hot backup of Oracle database in ApplicationHA environment	The hot backup of Oracle database is enabled by default in ApplicationHA environment. A virtual machine can fail during a hot backup of an Oracle database. ApplicationHA attempts to restart only if the following requirements are met: ■ The AutoEndBkup attribute value must be set to 1, which is the default. ■ The Startup option of Oracle agent must be STARTUP, STARTUP_FORCE, or CUSTOM. If you do not meet ApplicationHA requirements, you must manually end the hot backup. **Note:** If you set the AutoEndBkup attribute value to 0, then to avoid unexpected ApplicationHA behavior you must set the DetailMonitor attribute value to 1.

Note: If you want to modify or tune a particular attribute, use CLI or VOM.

Chapter 3

Configuring application monitoring with Symantec ApplicationHA

This chapter includes the following topics:

- About configuring application monitoring with ApplicationHA
- Before configuring application monitoring for Oracle
- Accessing the Symantec High Availability view
- Configuring application monitoring for Oracle

About configuring application monitoring with ApplicationHA

This chapter describes the steps to configure application monitoring with ApplicationHA in a virtualization environment.

Consider the following points before you proceed:

- You configure an application for monitoring on a virtual machine using the Symantec ApplicationHA Configuration Wizard.
- The Symantec ApplicationHA Configuration Wizard is launched when you click **Configure Application Monitoring** in the Symantec High Availability view of the Veritas Operations Manager (VOM) Management Server console.
- In this release, the wizard allows you to configure monitoring for only one application per virtual machine.

To configure another application using the wizard, you must first unconfigure the existing application monitoring.

- After you have configured monitoring for an application using the wizard, you can configure monitoring for other applications residing in the same virtual machine, using Symantec Cluster Server (VCS) commands.
  For more information read the following technote:
  http://www.symantec.com/docs/TECH159846
- After configuring Oracle for monitoring, if you create another Oracle instance, these new components are not monitored as part of the existing configuration.
  In such a case, you must first unconfigure the existing configuration and then reconfigure the application using the wizard. You can then select all the instances and databases for monitoring.

Before configuring application monitoring for Oracle

Ensure that you complete the following tasks before configuring application monitoring for Oracle on a virtual machine:

- Install Veritas Operations Manager (VOM) Management Server. For more information on working with VOM, see the *Symantec ApplicationHA User's Guide*. For information on accessing the Symantec High Availability view: See "Accessing the Symantec High Availability view " on page 28.
- Install ApplicationHA guest components on the virtual machine that you need to monitor.
- Assign ApplicationHA - Configure Application Monitoring (Admin) privileges to the logged-on user on the virtual machine where you want to configure application monitoring.
- Install the application and the associated components that you wish to monitor on the virtual machine.
- If you have configured a firewall, ensure that your firewall settings allow access to ports used by ApplicationHA installer, wizards, and services.
  Refer to the *Symantec ApplicationHA Installation Guide* for a list of ports and services used.

Accessing the Symantec High Availability view

To administer an application on a virtual machine that is running in the KVM environment, you must access the Symantec High Availability view of the Veritas Operations Manager (VOM) Management Server console.

From the Symantec High Availability view, you can perform administrative actions such as:

- Start an application
- Stop an application
- Configure application monitoring
- Unconfigure application monitoring
- Enable application heartbeat
- Disable application heartbeat
- Enter maintenance mode
- Exit maintenance mode

To access the Symantec High Availability view

1. Log on to the VOM Management Server console.
2. Select the Server perspective and expand Manage in the left pane.
3. Expand the Organization, or Uncategorized Hosts to navigate to the virtual machine.
4. Right-click the required virtual machine, and then click **Manage ApplicationHA**.

   The Symantec High Availability view appears.

Configuring application monitoring for Oracle

Perform the following steps to configure monitoring for Oracle on a virtual machine.

To configure application monitoring for Oracle

1. In the Symantec High Availability view of the VOM Management Server console, click **Configure Application Monitoring**.

   This launches the Symantec ApplicationHA Configuration Wizard.
2. Review the information on the Welcome screen and then click **Next**.

   The wizard lists all the supported applications for the system.
3. On the Application Selection page, click **Oracle** in the Supported Applications list.

4 Ensure that the Oracle database instances and associated Listener processes are running and then click **Next**.

An application discovery page appears.

Note: If the Oracle database instances are not running, ensure that they are added in the `oratab` file.

---

5 On Listener Selection panel, review the listed Oracle Listeners.

Listener	Name of the Listener.
Home	The $ORACLE_HOME path to Oracle binaries and configuration files.
Owner	The Oracle user, as the defined owner of executables and database files in /etc/passwd.
EnvFile	Specify the full path name of the file that is sourced by any operation.
TnsAdmin	The $TNS_ADMIN path to directory in which the Listener configuration file resides (listener.ora).
Password	This password is set for the selected Listener.

6 On the Oracle Database Selection screen, select the Oracle database instances and the associated listeners that you want to monitor and then click **Configure**.

Instances	Click to select the Oracle instances to monitor.
Database SID	The Oracle System ID (SID) is used to uniquely identify a particular database on a system. For this reason, one cannot have more than one database with the same SID on a computer system.
Oracle Home	ORACLE_HOME refers to a directory where the Oracle is installed
Database Owner	The Database Owner refers to the owner of the Oracle Home directory.
PFILE Path	PFILE is a text file that contains parameters and their values which are consumed by the Oracle system .
Listeners	Oracle Listener receives incoming client connection requests and manages the traffic of these requests to the database server.

Note: For each selected SID, all discovered Listeners are listed, and associated listeners are selected. If discovery phase is not displayed the correct SID - Listener association, you can appropriately select listeners for a selected SID.

7 The wizard performs the application monitoring configuration tasks. The ApplicationHA Configuration screen displays the status of each task.

After all the tasks are complete, click **Next**.

Note: If the configuration tasks fail, click **Diagnostic information** to check the details of the failure.

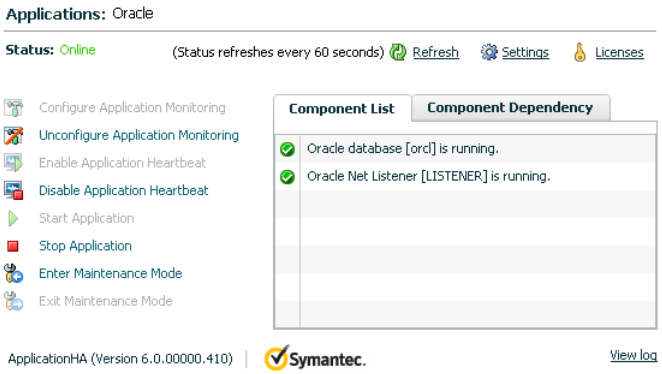
You then have to run the wizard again to configure the application monitoring.

8 Click **Finish** to complete the wizard.

This completes the application monitoring configuration.

9 To view the status of the configured application on a virtual machine, on the VOM Management Server console, right-click the appropriate virtual machine and then click **Manage ApplicationHA**.

The Symantec High Availability view appears.

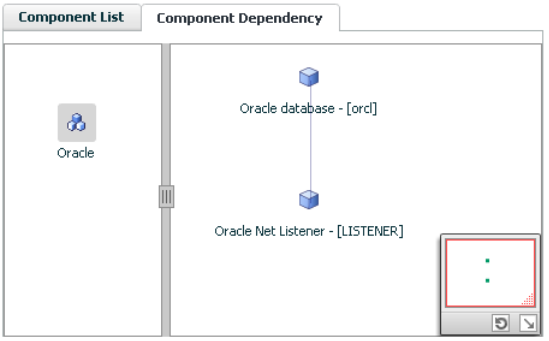


By default, the Component List tab appears. The tab lists each component of the configured application and the status description of each component.

For more information on viewing and administering applications through the Veritas Operations Manager, see the *Symantec ApplicationHA User's Guide*.

10 To view component dependency for the monitored application, click the **Component Dependency** tab.

The component dependency graph appears.



The graph illustrates the dependencies between a selected component group (an application or a group of inter-related components) and its components for the configured application. The left pane displays component groups and/or configured applications. The right pane displays components of the selected component group or application.

For more information on viewing component dependency for any configured application, see the *Symantec ApplicationHA User's Guide*.

Chapter 4

Troubleshooting Symantec ApplicationHA agent for Oracle

This chapter includes the following topics:

- About troubleshooting ApplicationHA agent for Oracle
- Error messages common to the Oracle and Netlsnr agents
- Error messages specific to the Oracle agent
- Error messages specific to the Netlsnr agent
- Verifying the Oracle health check binaries and intentional offline for an instance of Oracle

About troubleshooting ApplicationHA agent for Oracle

Review the information on the error logs that you must access:

- To check the Oracle installation error log, you must access:

  ```
  $ORACLE_BASE/oraInventory/logs/installActionsdate_time.log
  ```

  This file contains the errors that occurred during installation. It clarifies the nature of the error and at exactly which point it occurred during the installation. If there are any installation problems, you must send this file to Tech Support for debugging the issue.

- To check the ApplicationHA log file, you must access:

```
/var/VRTSvcs/log/engine_A.log
/var/VRTSvcs/log/Oracle_A.log
/var/VRTSvcs/log/Netlsnr_A.log
```

These files contain all the actions that the ApplicationHA engine and other agents for Oracle perform.

Error messages common to the Oracle and Netlsnr agents

Table 4-1 lists the ApplicationHA agent for Oracle error messages with the description and a recommended solution, if available.

Note: %s in the following table must be replaced by the actual value applicable to the particular Oracle or Netlsnr instance configured in ApplicationHA.

Table 4-1 ApplicationHA agent for Oracle error messages

Message	Description and solution
No ORACLE_HOME specified	The Home attribute in the Oracle or Netlsnr type has not been set. Solution: Set the Home attribute value to the correct full path name of the Oracle home.
Oracle home directory %s does not exist	The string that is specified for the Home attribute in the Oracle or Netlsnr type is incorrect. Solution: Set the Home attribute value to the correct full path name of the Oracle home.
File %s is not a valid text file	The file that the EnvFile attribute specifies for sourcing the environment variables is not present, not readable, or is not a text file. Solution: Set the EnvFile attribute value to the correct full path name. Ensure that the file format is valid.
VCSAgExec returned failure when trying to execute in-depth test	Internal error. Solution: Contact Technical Support for further assistance.

Table 4-1 ApplicationHA agent for Oracle error messages *(continued)*

Message	Description and solution
Unable to open pipe from %s	Internal error. Solution: Contact Technical Support for further assistance.
Process %s restarted	Warning message to indicate that the PID for the Oracle process that is specified is different than the one registered by the previous monitor cycle.
Monitor procedure %s returned %s	MonScript failed to execute correctly. Solution: Debug MonScript to assess the exact cause of failure.
Monitor procedure %s did not exit, return value is %s	Internal error while executing MonScript. Solution: Contact Technical Support for further assistance.
No owner for Oracle executables was specified	The Owner attribute in the Oracle type has not been set. Solution: Set the Owner attribute value to the correct owner of the database binaries.
Invalid owner %s for Oracle executables was specified	The Operating System user that the Owner attribute specifies is invalid. Solution: Set the Owner attribute value to the correct owner of the database binaries.
Access to Monscript %s denied. Detail Monitoring will not be enabled!! Please specify a valid file.	The file that the MonScript attribute specifies is not accessible or not found. Solution: Make sure that the file name indicates a valid and accessible file.
Encountered errors while decrypting password!	The agent cannot decrypt the password you specified. Solution: Use vcsencrypt utility to create a new encrypted password and supply the password.

Error messages specific to the Oracle agent

Table 4-2 lists the error messages for the ApplicationHA agent for Oracle with the description and a recommended solution, if available.

Note: %s in the following table must be replaced by the actual Oracle database instance name configured in ApplicationHA.

Table 4-2 Oracle agent error messages

Message	Description and solution
No SID specified	The Sid attribute in the Oracle type has not been set. Solution: Set the Sid attribute value to the correct database instance.
sqlplus not found in %s/bin	The client utility svrmgrl is not found in the $ORACLE_HOME/bin directory. Solution: Verify that the Oracle home has been correctly specified and that this executable is present.
srvctl not found in %s/bin	The client utility srvctl is not found in the $ORACLE_HOME/bin directory. Solution: Verify that the Oracle home has been correctly specified and that this executable is present.
Oracle %s failed to stop	Warning message to indicate that the following commands were not successful in closing the Oracle instance in the clean or offline entry point: ■ Shutdown immediate ■ Shutdown abort
Oracle database %s not running	Warning message to indicate that the database instance was not running even before the clean or offline entry points were executed. Solution: No action required.
Oracle (%s) kill TERM %s	Warning message to indicate that the Oracle processes would be signaled with SIGTERM. Solution: No action required.

Table 4-2 Oracle agent error messages *(continued)*

Message	Description and solution
Oracle (%s) kill KILL %s	Warning message to indicate that the Oracle processes would be signaled with SIGKILL. Solution: No action required.
Database in QUIESCING/QUIESCED mode	Warning message to indicate that database is in QUIESCING or QUIESCED mode.
Database in RESTRICTED mode	Warning message to indicate that database is in RESTRICTED mode.
Database in SUSPENDED state	Warning message to indicate that database is in SUSPENDED state.
Resource %s - monitor procedure did not complete within the expected time.	Refer to Oracle's alert log for more information. When a monitor times out as many times as the value specified, the corresponding resource is brought down by calling the clean entry point. The resource is then marked FAULTED, or it is restarted, depending on the RestartLimit attribute value. Solution: Set the FaultOnMonitorTimeouts attribute value to 0 so that the monitor failures are not considered indicative of a resource fault. Another possible reason could be that automatic archiving was not enabled while setting up the database. Solution: Archive the database manually. If automatic archival is enabled, set the LOG_ARCHIVE_START parameter value in the file init.ora to TRUE.

Table 4-2 Oracle agent error messages *(continued)*

Message	Description and solution
Custom script /opt/VRTSagents/ha/bin/Oracle/start_custom.sql does not exist. Will not be able to start the database.	The agent could not find the custom script at the specified location to start the database. Solution: Make sure the custom file exists at the specified location and has valid permissions.
Custom script /opt/VRTSagents/ha/bin/Oracle/shut_custom.sql does not exist. Using default shutdown option.	The agent could not find the custom script at the specified location to stop the database. Solution: Make sure the custom file exists and the specified location and has valid permissions.
oraerror.dat did not have records that could be parsed	The file oraerror.dat is not present or has records in an unsupported format. Solution: Make sure the file exists and has data in the supported format.
Incorrect Monitor Option	The MonitorOption value is less than 0 or greater than 1. Solution: Set the MonitorOption attribute value to 0 or 1.
MonitorOption value not applicable for this Oracle Version	The health check monitoring option is selected when Oracle version is not Oracle 10g or later. Solution: Set the MonitorOption value to 0 to select the process check monitoring option.
VCSAgExec returned failure when trying to execute health check monitor test	Internal error. Solution: Contact Technical Support for further assistance.
VCSAgExec returned failure while trying to find Oracle version	Internal error. Solution: Contact Technical Support for further assistance.

Table 4-2 Oracle agent error messages *(continued)*

Message	Description and solution
One or more of the attributes User:Pword:Table:MonScript are not set correctly. Detail monitoring will not be enabled!! Unset the DetailMonitor attribute if you want to disable DetailMonitoring.	Detail Monitoring has been enabled but the necessary attributes for detail monitoring have not been set correctly. Solution: Set the values of the required attributes for detail monitoring correctly or set DetailMonitor attribute value to 0 to disable detail monitoring.

Error messages specific to the NetIsnr agent

Table 4-3 lists the NetIsnr agent error messages with the description and a recommended solution, if available.

Note: %s in the following table must be replaced by the actual NetIsnr instance name configured in ApplicationHA.

Table 4-3 NetIsnr agent error messages

Message	Description and solution
Cannot open process directory.	The agent could not process the /proc entries in the particular monitor cycle. Solution: No action required.
Listener process %s not running	Warning message to indicate that the Listener process was not running even before the clean or offline entry points were executed. Solution: No action required.
Listener %s kill TERM %s	Warning message to indicate that the Listener process would be signaled with SIGTERM. Solution: No action required.
Listener %s kill KILL %s	Warning message to indicate that the Listener process would be signaled with SIGKILL. Solution: No action required.

Table 4-3 Netlsnr agent error messages *(continued)*

Message	Description and solution
lsnrctl not found in %s/bin	The client utility lsnrctl is not found in the $ORACLE_HOME/bin directory. Solution: Verify that the Oracle home has been correctly specified and that this executable is present.
lsnrctl operation timed out	The tnslsnr process does not respond. Solution: Verify the underlying network protocol.

Verifying the Oracle health check binaries and intentional offline for an instance of Oracle

This section describes the steps to verify the state of Oracle instance, the Oracle health check binaries, and intentional offline behavior for an Oracle agent.

Note: The steps listed in the table below should be executed by the operating system user specified in Owner attribute.

Table 4-4 lists the checks you can verify with Oracle health check binaries.

Table 4-4 Oracle health check options

Verify	Solution

If the ORACLE_HOME variable is set.	Run the following command to verify that the *$ORACLE_HOME* variable is correctly set.
	`# echo $ORACLE_HOME`
	Set the ORACLE_HOME environment variable if it is not already set. For example, run the following command to set the ORACLE_HOME variable:
	`# export ORACLE_HOME=/u01/oraHome`
	where `/u01/oraHome` is the Oracle home directory path.

Table 4-4 Oracle health check options *(continued)*

Verify	Solution
If the SID for the Oracle instance is set.	Run the following command to verify that the *$ORACLE_SID* variable is correctly set. `# echo $ORACLE_SID` If required, run the `# export ORACLE_SID=db` command to set the *$ORACLE_SID* variable.
If the library path is set.	Run the following command to verify that the *$LD_LIBRARY_PATH* variable is correctly set. `# echo $LD_LIBRARY_PATH` If required, run the `# export LD_LIBRARY_PATH=$ORACLE_HOME/lib:$LD_LIBRARY_PATH` command to set the *$LD_LIBRARY_PATH* variable.

If the Oracle instance is online.	Run the following command to verify that the Oracle health check binary reports the status correctly:
	`# ./oraapi_<Arch><Oracle_Version>`
	where *<Arch><Oracle_Version>* is the system architecture and the Oracle version.
	For example, `# ./oraapi_6411g`, where oraapi_6411g is a binary built for Oracle version 11 on a 64-bit system.
	The system displays the following message:
	`Instance is online`
	Run `# echo $?`
	If the system displays 110 the instance is online and active.

Table 4-4 Oracle health check options *(continued)*

Verify	Solution

If the Oracle instance is offline.	Run the following command to verify that the Oracle health check binary reports the status correctly:
	`# ./oraapi_<Arch><Oracle_Version>`
	where *<Arch><Oracle_Version>* is the system architecture and the Oracle version.
	For example, `# ./oraapi_6411g`, where oraapi_6411g is a binary built for Oracle version 11 on a 64-bit system.
	The system displays the following message:
	`Failure: Instance Shutdown class: SOFT`
	`Proper Shutdown`
	Run `# echo $?`
	If the system displays 100 the instance is offline.

If the Oracle instance is abnormally terminated.	If any of the important Oracle process is killed (for example, ora_pmon_<sid>, ora_smon_<sid>), the Oracle instance is abnormally terminated. In such a case run the Oracle health check binary, and the system displays following message:
	For example, run `# ./oraapi_6411g`
	`Failure: Abnormal Termination`
	`class: SOFT`
	`Abnormal termination`
	Run `# echo $?`
	If the system displays 98 the termination of Oracle processes. The VCS resource fails over as Oracle termination was abnormal and not intentional.

Verifying the intentional offline behavior of the VCS Oracle agent

Perform the following steps to verify if VCS Oracle agent determines a graceful shutdown of a resource as intentional offline.

1 Bring the resource online.

```
# hares -online ORA_oraprod -sys system
```

Where *system* is the name of the system that has Oracle database installed.

2 Verify that `IntentionalOffline` and `MonitorOption` attributes are set to 1.

3 Stop the Oracle instance properly outside VCS control.

```
# su - oracle

# bash

# bash-3.00$ sqlplus "/ as sysdba"

The system displays the following message:
SQL> shutdown immediate
Database closed.
Database dismounted.
ORA-03113: end-of-file on communication channel
```

4 You can verify the log files to confirm if the shutdown was graceful.

```
Sample log message:
VCS INFO V-16-1-13470 Resource ORA_oraprod
(Owner: Unspecified, Group: ORA_PROD_Group) is offline on system.
(Intentional But NOT initiated by VCS)
```

Oracle agent has identified the Intentional offline for the resource.

Appendix A

Resource type definitions

This appendix includes the following topics:

- About the resource type and attribute definitions
- Resource type definition for the Oracle agent
- Resource type definition for the Netlsnr agent

About the resource type and attribute definitions

The resource type represents the configuration definition of the agent and specifies how the agent is defined in the configuration file. The attribute definitions describe the attributes associated with the agent. The required attributes describe the attributes that must be configured for the agent to function.

Resource type definition for the Oracle agent

The ApplicationHA agent for Oracle is represented by the Oracle resource type in ApplicationHA.

```
type Oracle (

    static str AgentDirectory = "/opt/VRTSagents/ha/bin/Oracle"

static keylist SupportedActions = { VRTS_GetInstanceName,
        VRTS_GetRunningServices, DBRestrict, DBUndoRestrict,
        DBResume, DBSuspend, DBTbspBackup,
        "home.vfd", "owner.vfd", "getid", "pfile.vfd" }
static str ArgList[] = { Sid, Owner, Home, Pfile, StartUpOpt,
        ShutDownOpt, EnvFile, AutoEndBkup,
        User, Pword, Table, MonScript, Encoding,
        MonitorOption }
```

```
    str Sid
    str Owner
    str Home
    str Pfile
    str StartUpOpt = STARTUP_FORCE
    str ShutDownOpt = IMMEDIATE
    str EnvFile
    boolean AutoEndBkup = 1
    str MonScript = "./bin/Oracle/SqlTest.pl"
    str User
    str Pword
    str Table
    str Encoding
    int MonitorOption = 0

    static boolean IntentionalOffline = 0

)
```

Attribute definition for the Oracle agent

Review the description of the Oracle agent attributes. The agent attributes are classified as required, optional, and internal.

Table A-1 lists the required attributes. You must assign values to the required attributes.

Table A-1 Required attributes for Oracle agent

Required attributes	Type and dimension	Definition
Sid	string-scalar	The variable $ORACLE_SID that represents the Oracle instance. The Sid is considered case-sensitive by the Oracle agent and by the Oracle database server.
Owner	string-scalar	The Oracle user, as the defined owner of executables and database files in /etc/passwd. The agent also supports LDAP users as Oracle user.

Table A-1 Required attributes for Oracle agent *(continued)*

Required attributes	Type and dimension	Definition
Home	string-scalar	The $ORACLE_HOME path to Oracle binaries and configuration files. For example, you could specify the path as /opt/ora_home. **Note:** Do not append a slash (/) at the end of the path.

Table A-2 lists the optional attributes for Oracle agent. You can configure the optional attributes if necessary.

Table A-2 Optional attributes for Oracle agent

Optional Attributes	Type and Dimension	Definition
StartUpOpt	string-scalar	Startup options for the Oracle instance. This attribute can take the following values: ■ STARTUP ■ STARTUP_FORCE ■ RESTRICTED ■ RECOVERDB ■ CUSTOM ■ SRVCTLSTART ■ SRVCTLSTART_RO Default is STARTUP_FORCE. See "Startup and shutdown options for the Oracle agent" on page 14.

Table A-2 Optional attributes for Oracle agent *(continued)*

Optional Attributes	Type and Dimension	Definition
ShutDownOpt	string-scalar	Shut down options for the Oracle instance. This attribute can take the following values: ■ IMMEDIATE ■ TRANSACTIONAL ■ CUSTOM ■ SRVCTLSTOP ■ SRVCTLSTOP_TRANSACT ■ SRVCTLSTOP_ABORT ■ SRVCTLSTOP_IMMEDIATE Default is IMMEDIATE. See "Startup and shutdown options for the Oracle agent" on page 14.
EnvFile	string-scalar	The full path name of the file that is sourced by the entry point scripts. This file contains the environment variables set by the user for the Oracle database server environment such as LD_LIBRARY_PATH, NLS_DATE_FORMAT, and so on. The syntax for the contents of the file depends on the login shell of Owner. File must be readable by Owner. The file must not contain any prompts for user input.
Pfile	string-scalar	The name of the initialization parameter file with the complete path of the startup profile. You can also use the server parameter file. Create a one-line text initialization parameter file that contains only the SPFILE parameter. See the Oracle documentation for more information. See "Using the SPFILE in an ApplicationHA virtual machine" on page 61.

Table A-2 Optional attributes for Oracle agent *(continued)*

Optional Attributes	Type and Dimension	Definition
AutoEndBkup	integer-scalar	Setting the AutoEndBkup attribute to a non-zero value takes the datafiles in the database out of the backup mode, during Online. Default = 1 See "About ApplicationHA requirements for installing Oracle" on page 24.
MonitorOption	integer-scalar	Monitor options for the Oracle instance. This attribute can take values 0 or 1. ■ 0—Process check monitoring ■ 1—Health check monitoring You must set the value of this attribute as 1 to use the intentional offline functionality of the agent. Default = 0 See "Monitor options for the Oracle agent" on page 16.

Table A-2 Optional attributes for Oracle agent *(continued)*

Optional Attributes	Type and Dimension	Definition
LevelTwoMonitorFreq	integer-scalar	This resource-type level attribute specifies the frequency at which the agent for this resource type must perform second-level or detailed monitoring. You can also override the value of this attribute at resource-level. The value indicates the number of monitor cycles after which the agent must perform second-level or detailed monitoring. For example, the value 5 indicates that the agent will perform detailed monitoring every five online monitor intervals. **Note:** Upgrade from Symantec ApplicationHA 5.1 SP2 to Symantec ApplicationHA 6.1 fails if the DetailMonitor attribute had been used for detailed monitoring. If you are upgrading to Symantec ApplicationHA 6.1, set the value of the DetailMonitor attribute to the LevelTwoMonitorFreq attribute and reset the value of the DetailMonitor attribute to 0. **Note:** If you set the AutoEndBkup attribute value to 0, then make sure that the LevelTwoMonitorFreq attribute value is set to 1 for detail monitoring. Default = 0
MonScript	string-scalar	Pathname to the script provided for detail monitoring. The default (basic monitoring) is to monitor the database PIDs only. **Note:** Detail monitoring is disabled if the value of the attribute MonScript is invalid or is set to an empty string. The path of the supplied detail monitor script is /opt/VRTSagents/ha/bin/Oracle/SqlTest.pl. MonScript also accepts a pathname relative to /opt/VRTSagents/ha. A relative pathname should start with "./", as in the path ./bin/Oracle/SqlTest.pl.

Table A-2 Optional attributes for Oracle agent *(continued)*

Optional Attributes	Type and Dimension	Definition
User	string-scalar	Internal database user. Connects to the database for detail monitoring.
Pword	string-scalar	Encrypted password for internal database-user authentication. Encrypt passwords only when entering them using the command-line. Passwords must be encrypted using the ApplicationHA Encrypt utility.
Table	string-scalar	Table for update by `User/Pword`.
IntentionalOffline	static-boolean	This resource-type level attribute defines how VCS reacts when Oracle is intentionally stopped outside of VCS control. If you stop Oracle out of VCS control, the agent behavior is as follows: ■ 0—The Oracle agent registers a fault and initiates the failover of the service group. ■ 1—The Oracle agent takes the Oracle resource offline when Health check monitoring is enabled. If Health check monitoring is not enabled, the agent registers a fault and initiates the failover of the service group. **Note:** If you want to use the intentional offline functionality of the agent, you must set the value of the MonitorOption attribute as 1 to enable Health check monitoring. Default = 0 See *Symantec ApplicationHA User's Guide*.

Table A-3 lists the internal attribute for Oracle agent. This attribute is for internal use only. Symantec recommends not to modify the value of this attribute.

Table A-3 Internal attributes for Oracle agent

Optional Attributes	Type and Dimension	Definition
AgentDirectory	static-string	Specifies the location of binaries, scripts, and other files related to the Oracle agent. Default is /opt/VRTSagents/ha/bin/Oracle.

Resource type definition for the Netlsnr agent

The Netlsnr agent of the ApplicationHA agent for Oracle is represented by the Netlsnr resource type in ApplicationHA.

```
type Netlsnr (

    static str AgentDirectory = "/opt/VRTSagents/ha/bin/Netlsnr"

    static keylist SupportedActions = { VRTS_GetInstanceName,
        VRTS_GetRunningServices, "tnsadmin.vfd" }
    static str ArgList[] = { Owner, Home, TnsAdmin, Listener,
        EnvFile, MonScript, LsnrPwd, Encoding }
    str Owner
    str Home
    str TnsAdmin
    str Listener
    str EnvFile
    str MonScript = "./bin/Netlsnr/LsnrTest.pl"
    str LsnrPwd
    str Encoding
    static boolean IntentionalOffline = 0

)
```

Attribute definition for the Netlsnr agent

Review the description of the Netlsnr agent attributes. The agent attributes are classified as required, optional, and internal.

Table A-4 lists the required attributes for Netlsnr agent. You must assign values to the required attributes.

Table A-4 Required attributes for NetIsnr agent

Required attributes	Type and dimension	Definition
Owner	string-scalar	The Oracle user, as the defined owner of executables and database files in /etc/passwd. The agent also supports LDAP users as Oracle user.
Home	string-scalar	The $ORACLE_HOME path to Oracle binaries and configuration files. For example, you could specify the path as /opt/ora_home. Do not append a slash (/) at the end of the path.

Table A-5 lists the optional attributes for NetIsnr agent. You can configure the optional attributes if necessary.

Table A-5 Optional attributes for NetIsnr agent

Optional attributes	Type and dimension	Definition
TnsAdmin	string-scalar	The $TNS_ADMIN path to directory in which the Listener configuration file resides (listener.ora). Default is /var/opt/oracle.
Listener	string-scalar	Name of Listener. The name for Listener is considered case-insensitive by the NetIsnr agent and the Oracle database server. Default is LISTENER.
LsnrPwd	string-scalar	The ApplicationHA encrypted password used to stop and monitor the listener. This password is set in the Listener configuration file. Encrypt passwords only when entering them using the command-line. Passwords must be encrypted using the ApplicationHA Encrypt utility.

Table A-5 Optional attributes for Netlsnr agent *(continued)*

Optional attributes	Type and dimension	Definition
EnvFile	string-scalar	Specifies the full path name of the file that is sourced by the entry point scripts. This file contains the environment variables set by the user for the Oracle listener environment such as LD_LIBRARY_PATH and so on. The syntax for the contents of the file depends on the login shell of Owner. This file must readable by Owner. The file must not contain any prompts for user input.
MonScript	string-scalar	Pathname to the script provided for detail monitoring. By default, the detail monitoring is enabled to monitor the listener process. **Note:** If the value of the attribute MonScript is set to an empty string, the agent disables detail monitoring. The pathname to the supplied detail monitoring script is /opt/VRTSagents/ha/bin/Netlsnr/LsnrTest.pl. MonScript also accepts a pathname relative to /opt/VRTSagents/ha. A relative pathname should start with "./", as in the path ./bin/Netlsnr/LsnrTest.pl.
Encoding	string-scalar	Specifies operating system encoding that corresponds to Oracle encoding for the displayed Oracle output. Default is "".
IntentionalOffline	static-boolean	Do not change the value of this attribute. Default = 0

Table A-6 lists the internal attribute for Netlsnr agent. This attribute is for internal use only. Symantec recommends not to modify the value of this attribute.

Note: To change the value of any attribute, use CLI /Veritas Operations Manager

Table A-6 Internal attributes for Netlsnr agent

Optional Attributes	Type and Dimension	Definition
AgentDirectory	static-string	Specifies the location of binaries, scripts, and other files related to the Netlsnr agent. Default is /opt/VRTSagents/ha/bin/Netlsnr.

Appendix B

Detail monitoring

This appendix includes the following topics:

- Setting the PATH variable
- Setting up detail monitoring for ApplicationHA agent for Oracle
- Setting up detail monitoring for ApplicationHA agent for Netlsnr

Setting the PATH variable

ApplicationHA commands reside in the `/opt/VRTS/bin` directory. Add this directory to your PATH environment variable.

To set the PATH variable

- Perform one of the following steps:

  For the Bourne Shell (sh or ksh), type:

  ```
  $ PATH=/opt/VRTS/bin:$PATH; export PATH
  ```

  For the C Shell (csh or tcsh), type:

  ```
  $ setenv PATH :/opt/VRTS/bin:$PATH
  ```

Setting up detail monitoring for ApplicationHA agent for Oracle

The ApplicationHA agent for Oracle provides two levels of application monitoring: primary (basic monitoring) and secondary (detail monitoring).

- In the basic monitoring mode, the agent monitors the Oracle processes to verify that they are continuously active.

- In the detail monitoring mode, the agent executes the script defined in the attribute MonScript of the Oracle and the Netlsnr resources. If the script successfully executes, the agent considers the resource available. You can customize the default script according to your configuration.

See "Monitor options for the Oracle agent" on page 16.

You can use the agent's detail monitoring capability to monitor the status of a database and listener and increase the confidence in their availability. Before setting up detail monitoring, you must have the agent running satisfactorily at the basic level of monitoring.

Note: Disable detail monitoring before undertaking any database maintenance that involves disabling database access to external users.

Setting up detail monitoring for Oracle

Detail monitoring for an Oracle resource verifies whether a database is ready for transactions by performing an update transaction against a table within the database. The update action is taken by the two scripts, `SqlTest.pl` and `SimpleTest.pl`, provided with the ApplicationHA agent for Oracle. The scripts are available under the directory /opt/VRTSagents/ha/bin/Oracle/. Both scripts update the timestamp to monitor the database.

The `SqlTest.pl` script checks whether the database is open before updating the timestamp. If the database is found to be in restricted mode, quiesced mode, or suspended mode, the monitor returns success. In such a case, only basic monitoring occurs. The `SimpleTest.pl` script does not perform database checks but only issues update statements against the table.

Before enabling detail monitoring for Oracle, you must create a test table (with a timestamp) in the Oracle database. The agent uses this test table for internal purposes. Symantec recommends that you do not perform any other transaction on the test table. The detail monitor script, `MonScript`, must exist and have execute permission for root. You can use a custom monitor script, or the scripts provided with the agent. In the monitor script, the return code 100 indicates failure. Return codes from 101 to 110 indicate success.

The example to set up detail monitoring, based on the use of the supplied script, shows how to create and test a table for use by detail monitoring, and how to enable detail monitoring.

To set up detail monitoring for Oracle

1 Make the ApplicationHA configuration writable:

```
haconf -makerw
```

2 Freeze the service group to avoid automated actions by ApplicationHA caused by an incomplete reconfiguration:

```
hagrp -freeze DiscoveredOracleSG
```

3 Log on as an Oracle user.

```
su - <Owner>
```

4 Set the environment variables for ORACLE_HOME and ORACLE_SID.

```
export ORACLE_HOME=<Home>
export ORACLE_SID=<Sid>
```

5 Start the `sqlplus` utility to set up a database table:

```
$ORACLE_HOME/bin/sqlplus /nolog
```

6 As the database administrator, issue the following statements at the `sqlplus` prompt to create the test table:

```
connect / as sysdba

create user  <User>
identified by <Pword>
default tablespace USERS

temporary tablespace TEMP
quota 100K on USERS;

grant create  session to <User>;

create table <User>.<Table> (  tstamp date );
insert into <User>.<Table> (tstamp) values  (SYSDATE);
```

7 To test the database table for use, do the following:

```
disconnect
connect  <User>/<Pword>
update <User>.<Table> set (  tstamp ) = SYSDATE;

select TO_CHAR(tstamp, 'MON DD, YYYY HH:MI:SS  AM')
from <User>.<Table>;
exit
```

8 Enable the detail monitoring for the Oracle resource using the following ApplicationHA commands:

```
hares -modify Oracle_<SID>_res User User
hares -modify Oracle_<SID>_res Pword Pword
hares -modify Oracle_<SID>_res Table Table
hares -modify Oracle_<SID>_res MonScript  "./bin/Oracle/SqlTest.pl"
hares -override Oracle_<SID>_res LevelTwoMonitorFreq
hares -modify Oracle_<SID>_res LevelTwoMonitorFreq 1

haconf -dump  -makero

hagrp -unfreeze DiscoveredOracleSG
```

Enabling and disabling detail monitoring for Oracle resource

Review the instructions to enable or disable detail monitoring.

To enable detail monitoring

- Set the LevelTwoMonitorFreq attribute to 1.

```
hares -modify Oracle_<SID>_res LevelTwoMonitorFreq 1
```

To disable detail monitoring

- Set the LevelTwoMonitorFreq attribute to 0.

```
hares -modify Oracle_<SID>_res LevelTwoMonitorFreq 0
```

Setting up detail monitoring for ApplicationHA agent for NetIsnr

For NetIsnr agent, the detail monitoring is enabled by default to monitor the listener process.

You can disable detail monitoring by setting the value of the attribute MonScript to an empty string.

You can enable detail monitoring for NetIsnr by specifying a value for the MonScript attribute. The example to set up detail monitoring uses the supplied monitor script for NetIsnr, /opt/VRTSagents/ha/bin/NetIsnr/LsnrTest.pl. The detail monitoring script for the NetIsnr resource uses the Listener command `lsnrctl status $Listener` to test the Listener process.

To disable detail monitoring for NetIsnr

- Disable detail monitoring by setting the MonScript attribute to an empty string:

```
haconf -makerw
hagrp -freeze DiscoveredOracleSG
hares -modify LSNR_$Listener_res MonScript ""
haconf -dump -makero
hagrp -unfreeze DiscoveredOracleSG
```

To set up detail monitoring for NetIsnr

1. Make the ApplicationHA configuration writable:

```
haconf -makerw
```

2. Freeze the service group to avoid automated actions by ApplicationHA caused by an incomplete reconfiguration:

```
hagrp -freeze DiscoveredOracleSG
```

3. Enable detail monitoring by entering the following commands:

```
hares -modify LSNR_$Listener_res MonScript "./bin/NetIsnr/LsnrTest.pl"
haconf -dump -makero
hagrp -unfreeze DiscoveredOracleSG
```

Appendix C

Using the SPFILE in an ApplicationHA system for Oracle

This appendix includes the following topics:

- Using the SPFILE in an ApplicationHA virtual machine

Using the SPFILE in an ApplicationHA virtual machine

When using the ApplicationHA agent for Oracle, you can start a database instance by specifying a PFILE. If you do not specify the PFILE, the database instance starts up using the default SPFILE.

The agent attribute Pfile must specify the location of the PFILE. If your configuration uses the SPFILE, the contents of the PFILE must specify the location of the SPFILE, which must be created from the PFILE.

Note: To specify PFILE for Oracle instance, use CLI / Veritas Operation Manager (VOM).

To create the SPFILE from a PFILE

- The SPFILE must be created from the PFILE. You must have the sysdba or the sysoper system privileges to create an SPFILE.

  You can run the following command to create the SPFILE:

  ```
  CREATE SPFILE [= spfile_name] FROM PFILE [= pfile_name ];
  ```

  If you do not specify the complete path for the SPFILE, this command creates an SPFILE at the default location ($ORACLE_HOME/dbs on Linux).

To specify the SPFILE location in the PFILE

- To specify the location of the SPFILE in a PFILE, create a PFILE and specify the following entry in the PFILE:

  ```
  SPFILE = spfile_location
  ```

  The variable *spfile_location* represents the complete path of the SPFILE. For example:

  ```
  SPFILE = /database/startup/spfileora1.ora
  ```

  In this case, to start the database use the following command:

  ```
  startup pfile=location_of_pfile
  ```

Appendix D

Best practices

This appendix includes the following topics:

- Best practices for multiple Oracle instance configurations in a ApplicationHA environment

Best practices for multiple Oracle instance configurations in a ApplicationHA environment

Review some of the best practices for using multiple Oracle instances in a ApplicationHA environment:

- Define the system parameters such that the allocation of semaphore and shared memory is appropriate on the virtual machine.
- Use a dedicated set of binaries for each Oracle instance, even if each instance uses the same Oracle version.
- If your configuration uses the same Oracle version for all instances, install a version on the root disk or preferably on a secondary disk. Locate the pfiles in the default location and define several listener processes.
- If your configuration has different versions of Oracle, create a separate $ORACLE_HOME for each Oracle version.
- Listeners accompanying different versions of Oracle may not be backward-compatible. So, if you want to create a single listener.ora file, you must verify that the listener supports the other versions of Oracle in the virtual machine. You must also create a separate Envfile for each version of Oracle.
- Make sure that each listener listens to a different virtual address. Also, assign different names to listeners and make sure that they do not listen to the same port.