Cluster Server 7.4.2 Agent for Oracle Installation and Configuration Guide - Solaris

VERITAS™

Last updated: 2020-06-15

Legal Notice

Copyright © 2020 Veritas Technologies LLC. All rights reserved.

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

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

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

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

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

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

Technical Support

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

You can manage your Veritas account information at the following URL:
https://my.veritas.com

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

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

Documentation

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

Documentation feedback

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

You can also see documentation information or ask a question on the Veritas community site:
http://www.veritas.com/community/

Veritas Services and Operations Readiness Tools (SORT)

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

Contents

Chapter 1 Introducing the Cluster Server agent for Oracle 9

About the Cluster Server agent for Oracle 9
About the agent for Oracle ASM 10
Supported software for VCS agent for Oracle 10
How the agent makes Oracle highly available 10
How the Oracle and Netlsnr agents support intelligent resource monitoring 11
How the agent monitors Oracle instances running in Solaris zones 12
How the agent makes Oracle ASM instance highly available 12
How the agent supports Oracle container databases (CDBs) and pluggable databases (PDBs) 13
About Cluster Server agent functions for Oracle 13
Oracle agent functions 14
Netlsnr agent functions 28
ASMInst agent functions 29
ASMDG agent functions 32
Typical Oracle configuration in a VCS cluster 33
About setting up Oracle in a VCS cluster 34

Chapter 2 Installing and configuring Oracle 36

About installing Oracle in a VCS environment 36
Before you install Oracle in a VCS environment 37
About VCS requirements for installing Oracle 37
Location of the $ORACLE_HOME 41
Replacing the long pathnames for $ORACLE_HOME in the agent attributes 42
Failing over Oracle after a VCS node failure during hot backup 43
About Oracle installation tasks for VCS 44
Installation tasks for $ORACLE_HOME on shared disks 45
Installation tasks for $ORACLE_HOME on local disks 46
Installation tasks for Oracle using ASM 46
Installing ASM binaries in a VCS environment 48

Configuring VCS to support Oracle Restart function 50
Configuring Oracle ASM on the first node of the cluster 51
Enabling the clustering daemon for ASM-managed database 52
Disabling automatic startup of ASM 52
Configuring and starting up ASM on remaining nodes 53
Installing Oracle binaries on the first node of the cluster 54
Disabling the clustering daemon 54
Configuring the Oracle database 55
Copying the $ORACLE_BASE/admin/SID directory 56
Copying the Oracle ASM initialization parameter file 56
Verifying access to the Oracle database 56

Chapter 3 Installing and removing the agent for Oracle 59

Before you install or upgrade the agent for Oracle 59
Installing the VCS agent for Oracle software 59
Upgrading the VCS agent for Oracle 62
Upgrading VCS agent for Oracle 62
Upgrading VCS 3.5 agent for Oracle to use with VCS 7.4.2 (Solaris SPARC) 64
Disabling the Cluster Server agent for Oracle 65
Removing the Cluster Server agent for Oracle 66

Chapter 4 Configuring VCS service groups for Oracle 67

About configuring a service group for Oracle 67
Configuring Oracle instances in VCS 68
Configuring a single Oracle instance in VCS 68
Configuring multiple Oracle instances (single listener) in VCS 70
Configuring multiple Oracle instances (multiple listeners) in VCS 71
Configuring an Oracle instance with shared server support in VCS 73
Before you configure the VCS service group for Oracle 75
Importing the type definition files for Cluster Server agent for Oracle 76
Configuring the VCS service group for Oracle 77
Configuring the VCS service group for Oracle using Cluster Manager (Java console) 78
Configuring the VCS service group for Oracle using the command-line 80
Setting up detail monitoring for VCS agents for Oracle 83

Setting up detail monitoring for Oracle 84
Setting up detail monitoring for Netlsnr 88
Enabling and disabling intelligent resource monitoring for agents manually 89
Administering the AMF kernel driver 91

Chapter 5 Administering VCS service groups for Oracle 93

About administering VCS service groups 93
Bringing the service group online 94
Taking the service group offline 94
Switching the service group 94
Modifying the service group configuration 95

Chapter 6 Pluggable database (PDB) migration 96

Migrating Pluggable Databases (PDB) between Container Databases (CDB) 96

Chapter 7 Troubleshooting Cluster Server agent for Oracle 101

About troubleshooting Cluster Server agent for Oracle 101
Error messages common to the Oracle and Netlsnr agents 102
Error messages specific to the Oracle agent 104
Error messages specific to the Netlsnr agent 108
Error messages specific to the ASMInst agent 109
Error messages specific to the ASMDG agent 109
Troubleshooting issues specific to Oracle in a VCS environment 110
Verifying the Oracle health check binaries and intentional offline for an instance of Oracle 111
Verifying the intentional offline behavior of the VCS Oracle agent 113
Disabling IMF for a PDB resource 114

Appendix A Resource type definitions 115

About the resource type and attribute definitions 115
Resource type definition for the Oracle agent 115
Attribute definitions for the Oracle agent 116
About the Sid attribute in a policy managed database 125
Resource type definition for the Netlsnr agent 126
Attribute definition for the Netlsnr agent 127

Resource type definition for the ASMInst agent 131
Attribute definition for the ASMInst agent 131
Resource type definition for the ASMDG agent 136
Attribute definition for the ASMDG agent 136

Appendix B Sample configurations 139

About the sample configurations for Oracle enterprise agent 139
Sample single Oracle instance configuration 140
Sample VCS configuration file for single Oracle instance 141
Sample multiple Oracle instances (single listener) configuration 143
Sample VCS configuration file for multiple Oracle instances (single listener) 145
Sample multiple instance (multiple listeners) configuration 150
Sample VCS configuration file for multiple Oracle instances (multiple listeners) 150
Sample Oracle configuration with shared server support 154
Sample VCS configuration file for Oracle instance configured with shared server support 155
Sample configuration for Oracle instances in Solaris zones 158
Zone root on local disk for single Oracle instance 158
Zone root on shared disk for single Oracle instance 159
Zone root on local disk for multiple Oracle instances 160
Zone root on shared disk for multiple Oracle instances 163
Zone root on local disk for Oracle instance with shared server support 165
Zone root on shared disk for Oracle instance with shared server support 166
Oracle instance in the context of project 167
Sample VCS configuration file for zone root on local disk or on shared disk 168
Sample Oracle ASM configurations 169
Sample configuration for ASM disks as raw disks 169
Sample configuration for ASM disks as VxVM volumes 173
Sample configuration for ASM disks as CVM volumes 177
Sample configuration of Oracle pluggable database (PDB) resource in main.cf 182
Sample configuration of migratable Oracle pluggable database (PDB) resource in main.cf 185
Sample Configuration of Oracle supported by systemD 191
Sample configuration of ASMInst supported by systemD 193

Appendix C Best practices 195

Best practices for multiple Oracle instance configurations in a VCS environment 195

Appendix D Using the SPFILE in a VCS cluster for Oracle 197

About the Oracle initialization parameter files 197
Starting an Oracle instance 197
Using the SPFILE in a VCS cluster 198

Appendix E OHASD in a single instance database environment 200

About OHASD in a single instance database environment 200
Configuring the Application agent to make OHASD highly available 200

Chapter 1

Introducing the Cluster Server agent for Oracle

This chapter includes the following topics:

- About the Cluster Server agent for Oracle
- About the agent for Oracle ASM
- Supported software for VCS agent for Oracle
- How the agent makes Oracle highly available
- About Cluster Server agent functions for Oracle
- Typical Oracle configuration in a VCS cluster
- About setting up Oracle in a VCS cluster

About the Cluster Server agent for Oracle

The Cluster Server agent for Oracle provides high availability for Oracle database software.

Veritas high availability 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 agent package for Oracle 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.

About the agent for Oracle ASM

The Cluster Server agent suite for Oracle provides high availability agents for the Oracle databases that use Automatic Storage Management (ASM). The agents for Oracle ASM monitor the Oracle ASM instance and ASM disk groups.

The VCS Enterprise Agent package also contains the following agents for Oracle ASM:

- The ASMInst agent monitors the Oracle ASM instance.
- The ASMDG agent monitors the Oracle ASM disk groups.

Supported software for VCS agent for Oracle

Refer to the *Cluster Server Release Notes* for software versions supported by VCS 7.4.2 agent for Oracle.

Refer to the support matrix at https://www.veritas.com/support/en_US/article.000126342.

How the agent makes Oracle highly available

The Cluster Server agent for Oracle continuously monitors the Oracle database and listener processes to verify they function properly.

See "About Cluster Server agent functions for Oracle" on page 13.

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 or stopped states.
  The Oracle agent provides functionality to detect whether the Oracle resource was intentionally taken offline. The agent detects graceful shutdown for Oracle databases. 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 the value of the resource-level attribute 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.
- When the Oracle database is in READ WRITE mode, the agent performs an UPDATE query and when it is in READ ONLY mode, the agent runs SELECT query against the database.

The Oracle agent also supports IMF (Intelligent Monitoring Framework) in the process check mode of basic monitoring. IMF enables intelligent resource monitoring. The Oracle agent is IMF-aware and uses asynchronous monitoring framework (AMF) kernel driver for resource state change notifications.

See "How the Oracle and Netlsnr agents support intelligent resource monitoring" on page 11.

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 Oracle service group fails over to another node in the cluster. Thus the agent ensures high availability for the Oracle services and the database.

VCS provides high availability to applications that run in the context of Solaris Containers. You can configure the Cluster Server agent for Oracle to monitor these resources that run in the context of Solaris zones and projects.

See "How the agent monitors Oracle instances running in Solaris zones" on page 12.

How the Oracle and Netlsnr agents support intelligent resource monitoring

With intelligent monitoring framework (IMF), VCS supports intelligent resource monitoring in addition to the poll-based monitoring. Poll-based monitoring polls the resources periodically whereas intelligent monitoring performs asynchronous monitoring. You can enable or disable the intelligent resource monitoring functionality for the Oracle and Netlsnr agents.

See "Enabling and disabling intelligent resource monitoring for agents manually" on page 89.

When an IMF-enabled agent starts up, the agent initializes the asynchronous monitoring framework (AMF) kernel driver. After the resource is in a steady state, the agent registers the details that are required to monitor the resource with the AMF kernel driver. For example, the Oracle agent registers the PIDs of the Oracle processes with the AMF kernel driver. The agent's *imf_getnotification* function waits for any resource state changes. When the AMF kernel driver module notifies the *imf_getnotification* function about a resource state change, the agent framework runs the monitor agent function to ascertain the state of that resource. The agent notifies the state change to VCS which takes appropriate action.

See the *Cluster Server Administrator's Guide* for more information.

How the agent monitors Oracle instances running in Solaris zones

Solaris 11 provides a means of virtualizing operating system services, allowing one or more processes to run in isolation from other activity on the system. Such a "sandbox" is called a "non-global zone." Each zone can provide a rich and customized set of services. The processes that run in a “global zone” have the same set of privileges that are available on a Solaris system today.

VCS provides high availability to applications running in non-global zones by extending the failover capability to zones. VCS is installed in a global zone, and all the VCS agents and the engine components run in the global zone. For applications running within non-global zones, agents run script entry points inside the zones. If a zone configured under VCS control faults, VCS fails over the entire service group containing the zone.

See *Cluster Server Administrator's Guide*.

The Cluster Server agent for Oracle is zone-aware and can monitor Oracle instances running in non-global zones.

How the agent makes Oracle ASM instance highly available

Oracle provides ASM feature to store and manage the Oracle data that includes data files, control files, and log files. These Oracle data files that are stored in ASM disk groups are called ASM files.

For ASM-managed databases, you must start an ASM instance before you start the database instance. The ASM instance mounts ASM disk groups and makes ASM files available to database instances.

ASM requires Cluster Synchronization Services to enable synchronization between an ASM instance and a database instance.

See Oracle documentation.

The Cluster Server agent for Oracle has the following agents to keep the Oracle ASM instance and ASM disk groups highly available:

- ASMInst agent
  See "ASMInst agent functions" on page 29.
- ASMDG agent
  See "ASMDG agent functions" on page 32.

The Veritas high availability agent for Oracle continuously monitors the ASM instance and ASM disk groups to verify they function properly.

The Flex ASM functionality of Oracle is supported through the InfoScale ASMDG agent for Oracle.

The agents use `sysasm` role to connect to the ASM instances.

How the agent supports Oracle container databases (CDBs) and pluggable databases (PDBs)

The VCS agent for Oracle supports management of container and pluggable databases.

During FAULT of a CDB or critical PDBs, VCS fails over the entire container database from one server to another server along with the configured Pluggable databases. The PDBs, which are part of a CDB, are managed using separate PDB resources.

Note: Oracle agent does not support IMF, health check monitoring and intentional offline or PDBs.

See "Disabling IMF for a PDB resource" on page 114.

See "Migrating Pluggable Databases (PDB) between Container Databases (CDB)" on page 96.

About Cluster Server agent functions for Oracle

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

- Oracle agent functions
  See "Oracle agent functions" on page 14.
- Netlsnr agent functions
  See "Netlsnr agent functions" on page 28.

- ASMInst agent functions
  See "ASMInst agent functions" on page 29.
- ASMDG agent functions
  See "ASMDG agent functions" on page 32.

Oracle agent functions

The Oracle agent monitors the database processes.

Table 1-1 lists the Oracle agent functions.

Table 1-1 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 15.
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 15.
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 in traditional database and container database" on page 21.
oracle_imf_init	Initializes the agent to interface with the AMF kernel driver, which is the IMF notification module for Oracle agent. This function runs when the agent starts up.

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

Agent operation	Description
oracle_imf_ getnotification	Gets notification about resource state changes. This function runs after the agent initializes with the AMF kernel module. This function continuously waits for notification and takes action on the resource upon notification.
oracle_imf_register	Registers or unregisters resource entities with the AMF kernel module. This function runs for each resource after the resource goes into steady state (online or offline).
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
Info	Provides the static and dynamic information about the state of the database. **Note:** This attribute is not applicable PDB resources. See "Info entry point for Cluster Server agent for Oracle" on page 24.
Action	Performs the predefined actions on a resource. See "Action entry point for Cluster Server agent for Oracle" on page 25.

Startup and shutdown options for the Oracle agent

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

Note:

In case of the CDB or traditional database types, the use of the SRVCTLSTART or SRVCTLSTART_RO startup options along with the UseSystemD attribute is not supported.

In case of the PDB database types, the use of the UseSystemD attribute is not supported.

In either case, if the UseSystemD attribute is set, the agent reports the resource as UNKOWN and logs an appropriate message accordingly.

Table 1-2 lists the startup options that the agent supports for traditional database and container database.

Table 1-2 Startup options for traditional database and container database

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.
STARTUP	Runs the command `startup pfile='pfile_name_with_absolute _path'` if the pfile is configured. For example: `startup pfile = '</dbs/pfilename>'` If the pfile is not configured, the agent runs `startup` without *pfile*.
RESTRICTED	Starts the database in the RESTRICTED mode.
RECOVERDB	Performs a database recovery on instance startup.

Table 1-2 Startup options for traditional database and container database *(continued)*

Option	Description

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. **Note:** This is applicable only when an Oracle grid infrastructure is configured.
SRVCTLSTART_RO	Uses the srvctl utility to start an instance of the database with the `-o 'read only'` option. **Note:** This is applicable only when an Oracle grid infrastructure is configured.

Table 1-3 lists the shutdown options that the agent supports for traditional database and container database.

Table 1-3 Shutdown options for traditional database and container database

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 1-3 Shutdown options for traditional database and container database *(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. **Note:** This is applicable only when an Oracle grid infrastructure is configured.
SRVCTLSTOP_TRANSACT	Uses the `srvctl` utility to stop an instance of the database with the `-o transactional` option. **Note:** This is applicable only when an Oracle grid infrastructure is configured.
SRVCTLSTOP_ABORT	Uses the `srvctl` utility to stop an instance of the database with the `-o abort.` option **Note:** This is applicable only when an Oracle grid infrastructure is configured.
SRVCTLSTOP_IMMEDIATE	Uses the `srvctl` utility to stop an instance of the database with the `-o immediate` option.

Startup and shutdown options for the pluggable database (PDB)

You can specify Startup and Shutdown options for the PDB resources that are configured.

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

Table 1-4 Startup options for PDB

Options	Description
STARTUP	Runs the command `STARTUP PLUGGABLE DATABASE <pdbname>;` if the pdbname is configured.
STARTUP_FORCE	Runs the command `STARTUP PLUGGABLE DATABASE <pdbname> FORCE;` if the pdbname is configured.
RESTRICTED	Runs the command `STARTUP PLUGGABLE DATABASE <pdbname> RESTRICT;` if the pdbname is configured.
OPEN_RO	Runs the command `STARTUP PLUGGABLE DATABASE <pdbname> OPEN READ ONLY;` if the pdbname is configured.

During offline, shutdown of individual PDB will be attempted by running `shutdown immediate`.

See "Recommended startup modes for pluggable database (PDB) based on container database (CDB) startup modes" on page 19.

See "Migrating Pluggable Databases (PDB) between Container Databases (CDB)" on page 96.

Recommended startup modes for pluggable database (PDB) based on container database (CDB) startup modes

You can specify the startup options for the PDB based on the values configured in the CDB.

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

Table 1-5 CDB Startup options

CDB startup mode	Corresponding PDB startup modes
SRVCTLSTART	When the CDB is set with the startup value as SRVCTLSTART then the PDB can be assigned with one of the following values: ■ STARTUP ■ STARTUP_FORCE ■ RESTRICTED ■ OPEN_RO
SRVCTLSTART_RO	When the CDB is assigned with the SRVCTLSTART_RO startup value then the PDB should be OPEN_RO.
STARTUP	When the CDB is set with the startup value as STARTUP then the PDB can be assigned with one of the following values: ■ STARTUP ■ STARTUP_FORCE ■ RESTRICTED ■ OPEN_RO
STARTUP_FORCE	When the CDB is set with the startup value as STARTUP_FORCE then the PDB can be assigned with one of the following values: ■ STARTUP ■ STARTUP_FORCE ■ RESTRICTED ■ OPEN_RO
RESTRICTED	When the CDB is assigned with the startup value as RESTRICTED then the PDB should be RESTRICTED.
RECOVERDB	When the CDB is set with the startup value as RECOVERDB then the PDB can be assigned with one of the following values: ■ STARTUP ■ STARTUP_FORCE ■ RESTRICTED ■ OPEN_RO

Table 1-5 CDB Startup options *(continued)*

CDB startup mode	Corresponding PDB startup modes
CUSTOM	The custom script should start the PDB as CUSTOM.

See "Sample configuration of Oracle pluggable database (PDB) resource in main.cf" on page 182.

See "Migrating Pluggable Databases (PDB) between Container Databases (CDB)" on page 96.

Monitor options for the Oracle agent in traditional database and container database

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_dbw0`, `ora_smon`, `ora_pmon`, `ora_lmon`, and `ora_lgwr` processes to verify that Oracle is running. In this mode, the agent also supports intelligent resource monitoring. See "How the Oracle and Netlsnr agents support intelligent resource monitoring" on page 11.

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

Option	Description
1	Health check The agent uses the Health Check APIs from Oracle to monitor the 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. See "How the agent makes Oracle highly available" on page 10. The agent does not support intelligent resource monitoring in this mode. See "How the Oracle agent supports health check monitoring" on page 26.

In the detail monitoring mode, the agent performs a transaction on a test table in the database to ensure that Oracle database functions properly. The agent uses this test table for internal purposes. Veritas recommends that you do not perform any other transaction on the test table.

See "How the agent handles Oracle error codes during detail monitoring" on page 22.

See "Setting up detail monitoring for VCS agents for Oracle" on page 83.

Monitor for the pluggable database

During monitor, the Oracle agent connects to the CDB and fetches the status of all PDBs that are part of the CDB.

When IntentionalOffline is set for a CDB and PDB resource, and CDB is shut outside VCS then the PDB resource reports as Offline and not FAULTED. To enable this functionality, set the IntentionalOffline attribute to 1 for the PDB resource.

Note: Pluggable database will be failed over along with the Container database. Individual Pluggable database cannot be failed over using VCS.

How the agent handles Oracle error codes during detail monitoring

The Cluster Server 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, the order in which the Cluster Server 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 Veritas 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. See the *Cluster Server Administrator's Guide* for more information about VCS notification. 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.
FAILOVER (Default)	Marks the resource state as OFFLINE. This faults the service group by default, which fails over to the next available system. If the file oraerror.dat is not available, the agent assumes this default behavior for every Oracle error encountered.

Table 1-7 Predefined agent actions for Oracle errors *(continued)*

Action	Description
NOFAILOVER	Freezes the service group temporarily and marks the resource state as OFFLINE. The agent also sends a notification if the Notifier resource is configured. This action is typically associated with the errors that are not system-specific. For example, if a database does not open from a node due to corrupt Oracle files, failing it over to another node does not help.

Info entry point for Cluster Server agent for Oracle

The Cluster Server agent for Oracle supports the Info entry point, which provides static and dynamic information about the state of the database.

To invoke the Info entry point, type the following command:

```
# hares -value resource ResourceInfo [system]\
 [-clus cluster | -localclus]
```

For the above command to be executed successfully, you must perform the following command:

- Execute the resource refreshinfo using the following command:
  `# hares -refreshinfo <res-name> -sys <sys-name>`

The entry point retrieves the following static information:

- Version
- InstanceNo
- InstanceName
- DatabaseName
- HostName
- StartupTime
- Parallel
- Thread
- InstanceRole

The entry point retrieves the following dynamic information:

- InstanceStatus
- Logins
- OpenMode
- LogMode
- ShutdownPending
- DatabaseStatus
- Shared Pool Percent free
- Buffer Hits Percent

You can add additional attributes by adding sql statements to the file /opt/VRTSagents/ha/bin/Oracle/resinfo.sql. For example:

```
select 'static:HostName:'||host_name from v$instance;
select 'dynamic:ShutdownPending:'||shutdown_pending from
v$instance;
```

The format of the selected record must be as follows:

```
attribute_type:userkey_name:userkey_value
```

The variable *attribute_type* can take the value static and/or dynamic.

Action entry point for Cluster Server agent for Oracle

The Cluster Server agent for Oracle supports the Action entry point, which enables you to perform predefined actions on a resource.

To perform an action on a resource, type the following command:

```
# hares -action res token [-actionargs arg1 ...] \
[-sys system] [-clus cluster]
```

You can also add custom actions for the agent.

For further information, refer to the *Cluster Server Agent Developer's Guide*.

See Table 1-9 on page 26. describes the agent's predefined virtual fire drill actions.

Table 1-8 describes the agent's predefined actions.

Table 1-8 Predefined agent actions

Action	Description
VRTS_GetInstanceName	Retrieves the name of the configured instance. You can use this option for the Oracle and the Netlsnr resources.
VRTS_GetRunningServices	Retrieves the list of processes that the agent monitors. You can use this option for the Oracle and the Netlsnr resources.
DBRestrict	Changes the database session to enable the RESTRICTED mode.
DBUndoRestrict	Changes the database session to disable the RESTRICTED mode.
DBSuspend	Suspends a database.
DBResume	Resumes a suspended database.
DBTbspBackup	Backs up a tablespace; `actionargs` contains name of the tablespace to be backed up.

Note: VRTS_GetRunningServices, DBRestrict, DBResume, DBSuspend, DBTbspBackup, and DBUndoRestrict action entry points are not supported for Oracle PDB resources.

Table 1-9 lists the virtual fire drill actions of the Cluster Server agent for Oracle lets you run infrastructure checks and fix specific errors.

Table 1-9 Predefined virtual fire drill actions

Virtual fire drill action	Description
getid (Oracle agent)	Verifies that the Oracle Owner exists on the node.
home.vfd (Oracle agent)	Verifies the following: ■ ORACLE_HOME is mounted on the node and corresponding entry is in the fstab. If the ORACLE_HOME is not mounted, the action entry point checks if any other resource has already mounted ORACLE_HOME. ■ Pfile is provided and it exists on the node. ■ Password file from $ORACLE_HOME/dbs/orapw[SID] is present.
owner.vfd (Oracle agent)	Verifies the uid and gid of the Oracle Owner attribute. Checks if uid and gid of Owner attribute is the same on the node where the Oracle resource is currently ONLINE.
pfile.vfd (Oracle agent)	Checks for the presence of pfile or spfile on the local disk. If both pfile and spfile are not present, the agent function exits. If the Oracle resource is online in the cluster, the agent function logs a message that the spfile must be on the shared storage because the Oracle resource is online.
tnsadmin.vfd (Netlsnr agent)	Checks if listener.ora file is present. If the listener.ora file is not present, it checks if ORACLE_HOME is mounted and displays appropriate messages.

How the Oracle agent supports health check monitoring

The 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.

You can use the build_oraapi.sh script to link the libraries that are required for Oracle health check monitoring. Veritas ships the build_oraapi.sh script along with the VCS agent for Oracle in the /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.

Considerations for running build_oraapi.sh

- Run the build_oraapi.sh script on each node of the cluster where you want to use Oracle health monitoring.
- Run the build_oraapi.sh script inside each local container (Solaris zone) 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.
- Veritas ships the build_oraapi.sh script instead of 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 Oracle agent executable using build_oraapi.sh

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
```

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 -oracle_home /u01/oraHome
```

where */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 system architecture (32 bit or 64 bit) available on that node, and the full Oracle version.

For example, `oraapi_64_121010` is a binary created with Oracle version 12 on a 64-bit system.

See "Verifying the Oracle health check binaries and intentional offline for an instance of Oracle " on page 111.

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.

The Netlsnr agent is IMF-aware.

See "How the Oracle and Netlsnr agents support intelligent resource monitoring" on page 11.

Table 1-10 lists the Netlsnr agent functions.

Table 1-10 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.

Table 1-10 Netlsnr agent functions *(continued)*

Agent operation	Description
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)
netlsnr_imf_init	Initializes the agent to interface with the AMF kernel driver, which is the IMF notification module for Netlsnr agent. This function runs when the agent starts up.
netlsnr_imf_ getnotification	Gets notification about resource state change. This function runs after the agent registers with the AMF kernel module. This function continuously waits for notification and takes action on the resource upon notification.
netlsnr_imf_register	Registers or unregisters resource entities with the AMF kernel module. This function runs for each resource after the resource goes into steady state (online or offline).
Clean	Scans the process table for `tnslsnr $LISTENER` and kills it.
Action	Performs the predefined actions on a resource. See "Action entry point for Cluster Server agent for Oracle" on page 25.

ASMInst agent functions

The ASMInst agent monitors the processes of ASM instance.

Table 1-11 lists the ASMInst agent operations.

Table 1-11 ASMInst agent operations

Agent operation	Description
Online	Starts the Oracle ASM instance by using the following SQL command: `startup nomount` The Online operation starts the ASM instance without mounting any disk groups.
Offline	Stops the Oracle ASM instance by using the following SQL command: `shutdown immediate`
Monitor	Verifies the status of the ASM instance. The ASMInst agent uses process check monitoring to monitor the ASM instance. In the process check monitoring mode, the agent scans the process table for the asm_pmon, asm_smon, asm_lgwr, asm_rbal, and asm_dbw0 processes to verify that the ASM instance is running. The agent also checks if the ocssd.bin process is running. The agent returns offline for the following conditions: ■ The process is not running. ■ The process is restarted.
Clean	Forcibly stops the Oracle ASM instance by using the following SQL command: `shutdown abort` If the process does not respond to the `shutdown` command, the agent kills the process using the `SIGTERM` or `SIGKILL` commands.

ASMInst agent Startup options

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

The following table lists the startup options that the agent supports.

Table 1-12 Startup options

Option	Description
STARTUP	The default startup option for ASM instance resource. Uses `sqlplus` to start the ASM instance with the `startup nomount` startup option. If this option is used, none of the ASM disk groups get mounted.
STARTUP_MOUNT	Uses `sqlplus` to start the ASM instance with the `startup mount` startup option . If this option is used, all the ASM disk groups get mounted.
STARTUP_OPEN	Uses `sqlplus` to start the ASM instance with the `startup open` startup option. If this option is used, all the ASM disk groups are mounted and volumes are enabled.
SRVCTLSTART	Uses `srvctl` to start the ASM instance with the `nomount` startup option. If this option is used, none of the ASM disk groups get mounted.
SRVCTLSTART_MOUNT	Uses `srvctl` to start the ASM instance with the `-o mount` startup option. If this option is used, all the disk groups get mounted.
SRVCTLSTART_OPEN	Uses `srvctl` to start the ASM instance with the `-o open` startup option. If this option is used, all the ASM disk groups are mounted and volumes are enabled.

Note: Veritas recommends to use STARTUP or SRVCTLSTART option for ASMInst resource.

ASMInst agent Shutdown options

You can specify the Shutdown option for the Oracle ASM instances that are configured.

Table 1-13 Shutdown options

Option	Description
SRVCTLSTOP	Uses the srvctl utility to stop an ASM instance.

By default the agent uses the `sqlplus` command to stop the Oracle ASM instance. To stop the ASM instance using `srvctl` utility set the option to `SRVCTLSTOP`.

See "Attribute definition for the ASMInst agent" on page 131.

ASMDG agent functions

The ASMDG agent mounts the ASM disk groups that the Oracle databases use, monitors the status, unmounts the ASM disk groups.

You must have specified the disk group names in the DiskGroups attribute of the ASMDG agent.

Table 1-14 lists the ASMDG agent operations.

Table 1-14 ASMDG agent operations

Agent operation	Description
Online	Mounts the specified Oracle ASM disk groups to an ASM instance by using the following SQL command: `alter diskgroup dg_name1, dg_name2 mount`
Offline	Unmounts the specified Oracle ASM disk groups from an ASM instance by using the following SQL command: `alter diskgroup dg_name1, dg_name2 dismount` **Note:** The following Oracle message appears in the VCS log when an ASM instance with no ASM disk groups mounted is shut down: ORA-15100: invalid or missing diskgroup name

Table 1-14 ASMDG agent operations *(continued)*

Agent operation	Description
Monitor	Verifies the status of the specified ASM disk groups. The disk groups can be in one of the following states: ■ mounted ■ dismounted ■ unknown ■ broken ■ connected If multiple ASM disk groups are configured for a resource, then the ASMDG agent returns the resource state considering the status of all the specified ASM disk groups. **Note:** In case of Flex ASM, the ASMDG agent honors the cardinality configured for the ASM Instances. If the ASM instance is down on a particular node it is reported as offline if the number of ASM instances equal to the cardinality value and it is reported as Faulted if the number of ASM instances falls below the cardinality value.
Clean	Forcibly unmounts the Oracle ASM disk groups by using the following SQL command: `alter diskgroup dg_name1, dg_name2 dismount force`

Typical Oracle configuration in a VCS cluster

A typical Oracle configuration in a VCS cluster has the following characteristics:

- VCS is configured in a two-node cluster.
- The Oracle data is installed on shared storage.
- The Oracle binaries are installed locally on both nodes or on shared disks.

  If you use Oracle ASM feature, then the characteristics are as follows:

  - The Oracle binaries are installed locally on both the nodes.
  - The database files are installed on the ASM disk groups.

  Review the typical cluster setup illustration for ASM-managed database.
  See Figure 1-2 on page 34.
- The Cluster Server agent for Oracle is installed on both nodes.
  For ASM, ASMInst and ASMDG agents are installed on both nodes.

Figure 1-1 depicts a configuration where Oracle binaries and data are installed completely on shared disks.

Figure 1-1 Oracle binaries and data on shared disks

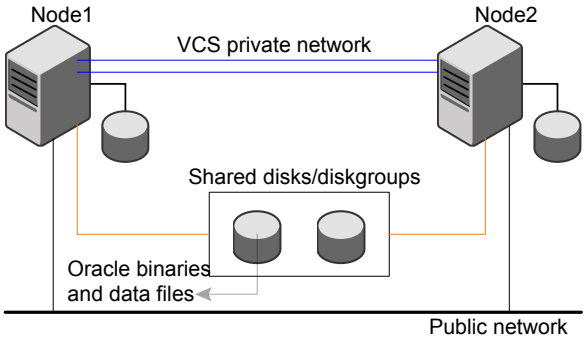
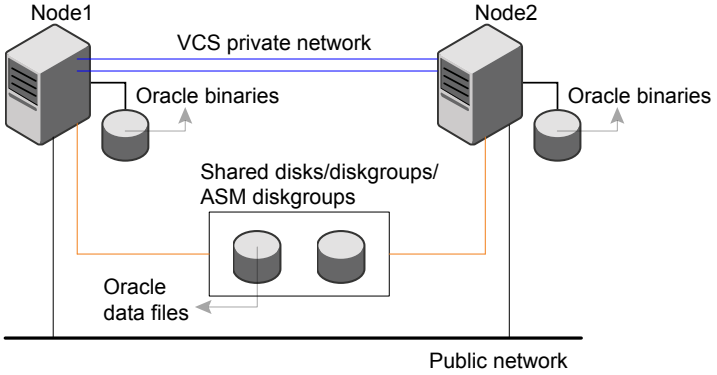


Figure 1-2 depicts a configuration where Oracle binaries are installed locally on each node and Oracle data is installed on shared disks.

Figure 1-2 Oracle binaries on local disk and Oracle data on shared disk



About setting up Oracle in a VCS cluster

Tasks involved in setting up Oracle in a VCS environment include:

- Setting up a VCS cluster
  Refer to *Cluster Server Installation Guide* for more information on installing and configuring VCS.
- Installing and configuring Oracle
  See "About installing Oracle in a VCS environment" on page 36.

- Installing the Cluster Server agent for Oracle
  See “Before you install or upgrade the agent for Oracle” on page 59.
- Configuring VCS service groups for Oracle
  See “About configuring a service group for Oracle” on page 67.

Chapter 2

Installing and configuring Oracle

This chapter includes the following topics:

- About installing Oracle in a VCS environment
- Before you install Oracle in a VCS environment
- About VCS requirements for installing Oracle
- About Oracle installation tasks for VCS
- Installing ASM binaries in a VCS environment
- Configuring Oracle ASM on the first node of the cluster
- Configuring and starting up ASM on remaining nodes
- Installing Oracle binaries on the first node of the cluster
- Configuring the Oracle database
- Copying the $ORACLE_BASE/admin/SID directory
- Copying the Oracle ASM initialization parameter file
- Verifying access to the Oracle database

About installing Oracle in a VCS environment

The strategy for installing Oracle into a VCS cluster is aimed at ensuring that installations on all nodes in the cluster are uniform.

See the Oracle documentation on Solaris.

You can install Oracle in the following ways in a VCS 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: To use ASM for Oracle database storage, you must install $ORACLE_HOME on the local disks of all nodes in the cluster.

See "About Oracle installation tasks for VCS" on page 44.

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

When installing Oracle, ensure that the login_id, id_name, group_id, and group_name for the Oracle owner is the same on all the nodes. The user oracle and the group dba must be local and not Network Information Service (NIS and NIS+) users.

Before you install Oracle in a VCS environment

Make sure you meet the following prerequisites:

- Verify that VCS is installed on all nodes in the cluster.
- Verify that all nodes in the cluster have adequate resources to run Oracle and VCS.
- Verify that the network supports the TCP/IP protocol.
- Make sure that you meet the VCS requirements to install Oracle.

About VCS requirements for installing Oracle

Make sure you meet the following requirements to install Oracle in a VCS cluster:

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.

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.

If you want to use Oracle ASM, then you must place the Oracle home directory only on the local disks of each node.

Review the advantages of each approach to make a decision.

See "Location of the $ORACLE_HOME" on page 41.

Configurations with multiple Oracle instances (SIDs)

You can have multiple Oracle instances that are defined in a single cluster. In such cases, the parameter file for each instance must be accessible on all the nodes in the service group's SystemList attribute.

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 (UFS 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

Location of core files for Oracle processes that terminate abnormally	The VCS agent framework sets the current directory to /opt/VRTSagents/ha/bin/Oracle before it runs the Oracle agent scripts or the programs that execute the Oracle binaries. Oracle binaries, which run as the user oracle, do not have permission to write to /opt/VRTSagents/ha/bin/Oracle. So, any "core" files that the Oracle binaries generate when the processes terminate abnormally are lost. Veritas recommends using the `coreadm` (1M) command on Solaris to specify the name and the location of such core files.
Transparent listener failover	You can enable Oracle Server clients to reconnect after a node switch without reconfiguring. For such reconnections you must include at least one IP resource in the service group for the Oracle resource. The hostname mapping the IP address of this resource must be used for the Host field in the file $TNS_ADMIN/listener.ora. If you use the TCP/IP protocol for Oracle client/server communication, verify that the file /etc/services contains the service name of the Oracle Net Service. You must verify this file on each node that is defined in the service group's SystemList attribute.
Listener authentication in VCS environment	The Netlsnr agent supports OS authentication as well as password authentication for the listener process. Veritas recommends that you 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. See "Encrypting Oracle database user and listener passwords" on page 82. Refer to the Oracle documentation for details on configuring the listener authentication.
Long pathname limitation for $ORACLE_HOME	The Solaris process table limits process pathnames to 79 characters. The full pathname of processes in $ORACLE_HOME can possibly have 80 characters or more. In this case, you can create a soft link to the $ORACLE_HOME directory. You can then use the soft link in place of the long filename in the Home attribute in the main.cf file. See "Replacing the long pathnames for $ORACLE_HOME in the agent attributes" on page 42.
Oracle NLS information	You can define the NLS information in one of the following ways: ■ Define the appropriate parameters in the Oracle parameter file. ■ Define the appropriate environment variables in the EnvFile attribute for the agents. See "About the resource type and attribute definitions" on page 115. Defining the parameters in the Oracle parameters file affects NLS settings for the Oracle server. Defining the environment variables affects the NLS input and output of client utilities.

Hot backup of Oracle database in VCS environment

The hot backup of Oracle database is enabled by default in VCS environment.

A node can fail during a hot backup of an Oracle database. During such failures, VCS can fail over to another node 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.
  See "Startup and shutdown options for the Oracle agent" on page 15.

If you do not meet VCS requirements, you must manually end the hot backup and then fail over Oracle to another node.

Note: If a node fails during a hot backup of container database or pluggable database, you must set AutoEndBkup attribute of the corresponding CDB resource to 1. When the AutoEndBkup is set to 1 for the CDB, it also ends the backup of both CDB and PDB during Online.

See "Failing over Oracle after a VCS node failure during hot backup" on page 43.

Note: If you set the AutoEndBkup attribute value to 0, then to avoid unexpected VCS behavior you must enable detail monitoring.

See "Setting up detail monitoring for VCS agents for Oracle" on page 83.

Storage devices for Oracle ASM configurations in VCS

You can choose one of the following storage devices for Oracle ASM:

- ASM disks as raw disks
  If you use raw disks, then make sure that the disks have the persistent permissions that are specified for ASM $ORACLE_HOME.
- ASM disks as Veritas Volume Manager volumes
  If you use VxVM volumes, then make sure that the disks have persistent permissions across reboots. The permissions must be the same as that of ASM $ORACLE_HOME.
  See Veritas Volume Manager documentation.
- ASM disks as Cluster Volume Manager volumes
  If you use CVM volumes, then make sure that the disks have persistent permissions across reboots. The permissions must be the same as that of ASM $ORACLE_HOME.
  See Veritas Volume Manager documentation.

If you want to configure mirroring for ASM disks that use VxVM or CVM volumes, then you must configure VxVM mirroring and not configure ASM mirroring.

See "Sample Oracle ASM configurations" on page 169.

The ASMInst agent does not support pfile or spfile for ASM instances on the ASM disk groups. Veritas recommends that you copy this file from ASM disk group to the local file system.

ASM instances configured on VxVM or CVM volumes in a Solaris zone environment

In a Solaris zone environment, you must do the following for the ASM instances that are configured on VxVM or CVM volumes:

- Create a new Oracle user in the global zone.
  The user name, group name, user id, and group id of the new Oracle user must be the same that in the local zone where you have configured ASM.
- To be able to perform I/O on the raw volumes, set the permissions as follows from the global zone:
  - 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:

  ```
  # 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
- Change the permissions of device matches for VxVM volumes in the local zone. For example:

  ```
  localzone# chown oracle:dba
  /dev/vx/rdsk/ASMDG/asmvol
  localzone# chmod 660
  /dev/vx/rdsk/ASMDG/asmvol
  ```

Location of the $ORACLE_HOME

You can place the Oracle home directory ($ORACLE_HOME), which contains the Oracle binaries and configuration files, locally on each server's disk. Alternatively, you can place the Oracle home directory on the shared storage. The correct location for Oracle binaries depends on your environment. The following points discuss the advantages of each approach.

$ORACLE_HOME directory on shared disks	You can install the Oracle Database Server ($ORACLE_HOME) on shared disks. Each node in the cluster must have the same mount point directory for the shared file system. Placing the Oracle binaries on shared storage simplifies setting up a given node in a cluster to run an instance. Each database service group is self-contained. An instance can be moved to a new node in the cluster that shares the storage. For example, in a cluster with four nodes, you can have three database instances or service groups, each at a different Oracle version. If the Oracle binaries are placed on shared storage, three copies of Oracle, that is, one per version are required on shared storage. By contrast, placing the Oracle binaries on local storage, would require as many as 12 copies of the binaries (three versions on four nodes). The disadvantage of this approach is that a rolling upgrade of Oracle binaries on shared storage is not possible.
$ORACLE_HOME directory on local disks	You can install the Oracle Database Server ($ORACLE_HOME) on the local disk. The advantage is that you can upgrade the Oracle database binaries on an offline node while the database server runs on another node. The database server can later be switched to the upgraded node (provided the database is compatible), permitting a minimum amount of downtime. The disadvantage of this approach is that with a large number of nodes, it becomes difficult to maintain the various Oracle installations.

If you want to use Oracle ASM, then you must place the Oracle home directory only on the local disks of each node. You can install Oracle ASM on the same Oracle home as Oracle Database, or you can install Oracle ASM on a different Oracle home directory.

See Oracle documentation.

Replacing the long pathnames for $ORACLE_HOME in the agent attributes

You must create the soft link for $ORACLE_HOME on each of the nodes in the cluster, and then modify the values for the agent attributes.

To replace the long pathnames

1. Create a soft link to the long pathname for $ORACLE_HOME. For example:

```
# ln -s /opt/apps/oracle/home/directory/is/longer/than\
/eighty/characters/oracle /opt/link_to_longpath
```

2. In the file /etc/VRTSvcs/conf/config/main.cf, modify the Home and the Pfile attributes for the Oracle and the Netlsnr resource types.

   For ASM, also modify these attributes for the ASMInst and the ASMDG resource types.

   See "About the sample configurations for Oracle enterprise agent" on page 139.

   For example:

```
Home = "/opt/link_to_longpath"
Pfile = "/opt/link_to_longpath/dbs/initVRT.ora"
```

Failing over Oracle after a VCS node failure during hot backup

If a node fails during a hot backup, VCS can succeed with a node failover only if you meet VCS requirements for Oracle hot backup.

See "About VCS requirements for installing Oracle" on page 37.

If you do not meet VCS requirements, the agent cannot open the database in the backup mode on the failover node. Therefore, VCS cannot online the Oracle group on the failover node. You may encounter the following Oracle errors in this situation:

```
$ ORA-1110 "data file %s: '%s'"
```

or

```
$ ORA-1113 "file %s needs media recovery"
```

In such cases, you must manually end the hot backup and then fail over Oracle to another node.

To manually fail over Oracle after a node failure that occurs during hot backup

1. Take the datafiles in the database out of the backup mode.

   Refer to the Oracle documentation for instructions on how to change the state of the database files.

2. Shut down the database.

3. Bring the Oracle service group online on the failover node.

   The agent can now start the Oracle instance, mount the database, and open the datafiles.

About Oracle installation tasks for VCS

Tasks to complete Oracle installation in a VCS cluster depend on whether you want the $ORACLE_HOME on shared disk or local disk.

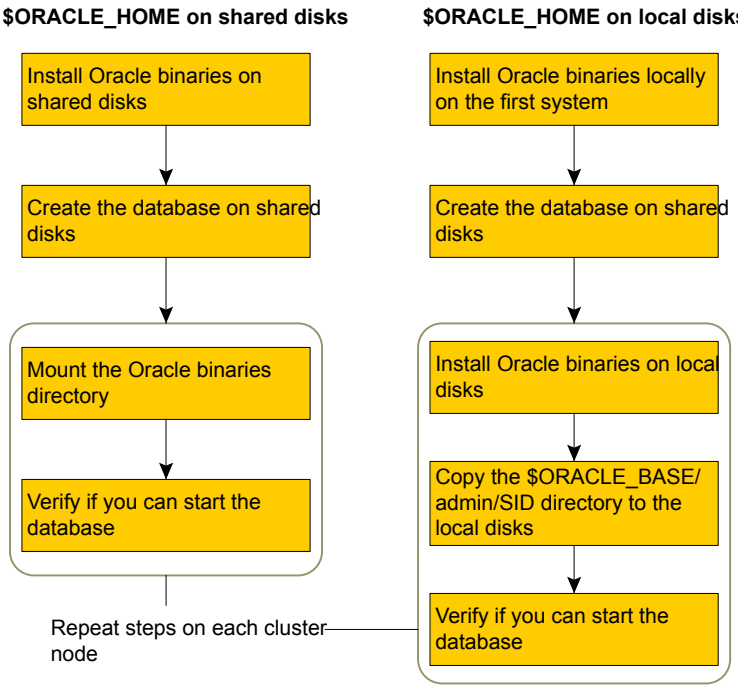
See "Installation tasks for $ORACLE_HOME on shared disks" on page 45.

See "Installation tasks for $ORACLE_HOME on local disks" on page 46.

For ASM, you must install $ORACLE_HOME on local disks and configure the Oracle ASM.

Figure 2-1 illustrates the flow of the installation when the $ORACLE_HOME is on shared disk and on local disk.

Figure 2-1 Comparison of the installation flow



Installation tasks for $ORACLE_HOME on shared disks

The methodology is to install the Oracle binaries and to create the Oracle database on shared disks during the first installation. Mount the Oracle binaries directory and verify that the database can be started from all nodes.

Table 2-1 lists the tasks to install Oracle such that $ORACLE_HOME is on shared disks.

Table 2-1 Installation tasks for $ORACLE_HOME on shared disks

Task	Reference
From any node in the cluster, install Oracle binaries on shared disks. Disable the Oracle clustering daemon.	See "Installing Oracle binaries on the first node of the cluster" on page 54. See "Disabling the clustering daemon" on page 54.
From the node where you installed Oracle, create the database on shared disks.	See "Configuring the Oracle database" on page 55.

Table 2-1 Installation tasks for $ORACLE_HOME on shared disks *(continued)*

Task	Reference
From each node that would be a part of the Oracle cluster, verify access to the database on shared disks.	See "Verifying access to the Oracle database" on page 56.

Installation tasks for $ORACLE_HOME on local disks

The methodology is to install the Oracle binaries on the local disk and to create the Oracle database on shared disks during the first installation. Then, install the Oracle binaries on local disks of other nodes. This methodology ensures that all Oracle installations are identical and access the database from the same location on the shared disk.

Table 2-2 lists the tasks to install Oracle such that $ORACLE_HOME is on local disks.

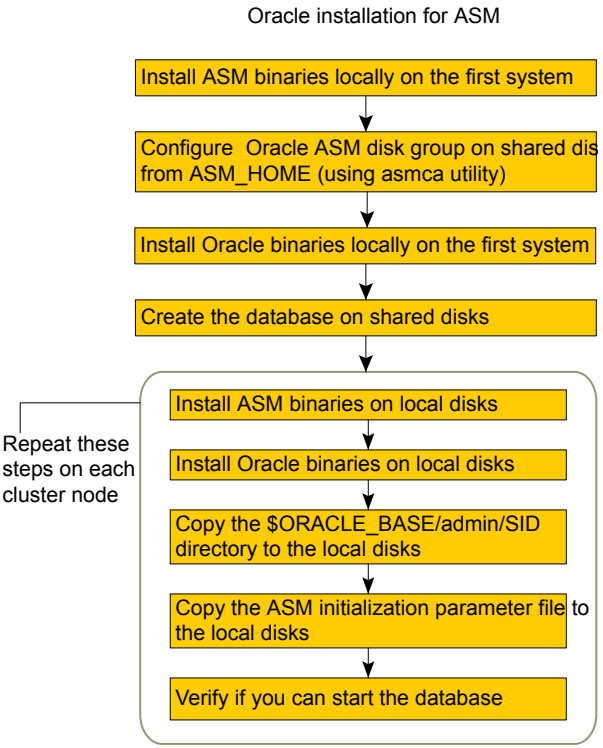
Table 2-2 Installation tasks for $ORACLE_HOME on local disks

Task	Reference
On the first node in the cluster, install Oracle binaries on the local disk.	See "Installing Oracle binaries on the first node of the cluster" on page 54.
From the first node where you installed Oracle, create the database on shared disks.	See "Configuring the Oracle database" on page 55.
On remaining nodes in the cluster, perform the following tasks: ■ Install Oracle binaries on the local disk. ■ Copy the $ORACLE_BASE/admin/SID directory from the first node to the remaining nodes. Copy $ORACLE_HOME/dbs/*SID* to the local disk. ■ Verify if you can start the database.	■ See "Installing Oracle binaries on the first node of the cluster" on page 54. ■ See "Copying the $ORACLE_BASE/admin/SID directory" on page 56. ■ See "Verifying access to the Oracle database" on page 56.

Installation tasks for Oracle using ASM

Figure 2-2 illustrates the flow of the installation when you want to use ASM for the Oracle database.

Figure 2-2 Installation flow for Oracle ASM



The methodology is to perform the following:

- Install the Oracle and the ASM binaries on the local disk, configure the ASM, and to create the Oracle database on ASM disk groups during the first installation.
- Then, install the Oracle and the ASM binaries on local disks of other nodes.

This methodology ensures that all Oracle installations are identical and access the database from the same location.

The following table lists the tasks to install Oracle and ASM such that $ORACLE_HOME is on local disks.

Table 2-3 Installation tasks for $ORACLE_HOME on local disks

Task	Reference
On the first node in the cluster, install ASM binaries on the local disk.	See "Installing ASM binaries in a VCS environment" on page 48.

Table 2-3 Installation tasks for $ORACLE_HOME on local disks *(continued)*

Task	Reference
From the first node where you installed ASM, configure Oracle ASM disk groups on shared disks (using `asmca` utility).	See "Configuring Oracle ASM on the first node of the cluster" on page 51.
On the first node in the cluster, install Oracle binaries on the local disk. Enable the Oracle clustering daemon.	See "Installing Oracle binaries on the first node of the cluster" on page 54. See "Enabling the clustering daemon for ASM-managed database" on page 52.
From the first node where you installed Oracle, create the database on ASM disk groups.	See "Configuring the Oracle database" on page 55.
On each node in the cluster, perform the following tasks: ■ Install ASM binaries on the local disk. ■ Install Oracle binaries on the local disk. ■ Copy the $ORACLE_BASE/admin/SID directory to the local disk. ■ Copy the ASM initialization parameter file to the local disk. ■ Verify if you can start the database.	See "Installing Oracle binaries on the first node of the cluster" on page 54. See "Copying the $ORACLE_BASE/admin/SID directory" on page 56. See "Copying the Oracle ASM initialization parameter file" on page 56. See "Verifying access to the Oracle database" on page 56.

Installing ASM binaries in a VCS environment

Perform the following procedure if want to configure Oracle ASM instances for high availability.

Note: If you want to use Oracle ASM feature, do not disable the Cluster Synchronization Services.

To install ASM binaries for Oracle for VCS

1. Run the Oracle installer as an Oracle user.
2. In the Select Installation Option dialog box, choose the **Install Grid Infrastructure Software Only** option and click **Next**.
3. Follow the wizard instructions and select other options.

4 In the Specify Installation Location dialog box, specify a location on the local disk for both Oracle base and Software location.

5 Follow the wizard instructions and select other options

6 In the Summary dialog box, review your selections. Click **Back** to make changes. Otherwise click **Finish**.

7 The installer prompts whether you want to instantiate a standalone version of the software or a cluster version of the software. Choose the version that you want to install.

- **Instantiate a standalone version**

Note: If you choose this option, you must execute `roothas.pl`. This option creates a configuration named **Oracle Restart**, which is used on standalone servers and not on RAC clusters. Therefore, you must perform this installation on all nodes in the VCS cluster that are used as failover destinations for Oracle Grid Infrastructure and Oracle ASM.

This option requires you to install the Oracle Grid Infrastructure software locally on every node. You must not use the same set of binaries on more than one node because the same CSSD daemon (used in a particular configuration) cannot be run on nodes with different host names, requiring you to reconfigure the daemon each time it fails over or is used on another node.

In the Oracle Restart configuration the newly created database instance and netlsnr resources automatically get registered to the Oracle Grid Infrastructure. In VCS, these resources are configured as part of failover service group.
If the database and netlsnr resources accidentally go offline, Oracle Grid Infrastructure tries to restart the resources. However, VCS tries to online these resources on another node within the failover service group. This behavior causes concurrency violation.

To avoid concurrency violation perform one of the following:

- In the Oracle Grid Infrastructure, set the auto_start attribute of each database and netlistner resource to 0.
- In VCS, set the RestartLimit attribute value of each database and netlistner resource to the corresponding value set for these resources in the RESTART_ATTEMPTS attribute in Oracle Grid Infrastructure. For detailed description on configuring VCS to support the Oracle retsart funtion refer to See "Configuring VCS to support Oracle Restart function" on page 50.

- **Don't instantiate a standalone version**
  Choose this option to instantiate a cluster version of the software. This is required for a RAC database.

8 Click **Next** after the installation is complete.

9 In the End of Installation dialog box, click **Exit**.

10 Proceed to configure ASM disk group on shared disks using the `asmca` utility.

See "Configuring Oracle ASM on the first node of the cluster" on page 51.

Configuring VCS to support Oracle Restart function

To ensure that VCS supports the Oracle Restart feature, the user needs to tune the VCS parameters so that it does not attempt a failover action when the Oracle grid infrastructure attempts to restart the resources on the same node. For VCS to take correct decision, the ToleranceLimit resource attribute needs to be tuned accordingly.

Oracle Restart feature uses the CHECK_INTERVAL and RESTART_ATTEMPTS attributes to determine the monitoring interval and restart attempts if the resources are unavailable.

For example, when ASM instance becomes unavailable on a node, the grid infrastructure tries to restart the ASM resource, however VCS detects that the application is unavailable and reports the resource as faulted and tries to failover the service group to another cluster node. To avoid such scenarios, we need to ensure that the VCS policies do not take effect until Oracle completes performing its functions.

You need to modify the VCS resource ToleranceLimit attribute such that ToleranceLimit * MonitorInterval is greater than (CHECK_INTERVAL *RESTART_ATTEMPTS) + MonitorInterval .

If Oracle's attribute values for ASM resource is CHECK_INTERVAL = 1 (in seconds) and RESTART_ATTEMPTS = 5 (no of restart attempts), and the VCS configuration values for the ASMInst resource has MonitorInterval set to 60, then set the ToleranceLimit to 2 based on the below calculations,

```
2 (ToleranceLimit) * 60 (MonitorInterval) = 120
1 (CHECK_INTERVAL) * 5 (RESTART_ATTEMPTS) + 60 (MonitorInterval) = 65
```

The following commands can be used to set the Attribute ToleranceLimit :

- `# hares -override <resource name> ToleranceLimit`
- # hares –modify *<resource name>* ToleranceLimit 2

Note: Tune the ToleranceLimit value for Oracle applications such as ASM, ASM Diskgroup, Oracle Database, and Oracle netlistner, that are configured with both Oracle Restart feature and VCS.

Configuring Oracle ASM on the first node of the cluster

If you want to use the ASM feature of Oracle, configure the Oracle ASM using the ASM Configuration Assistant.

You need to configure the ASM only once, from the node on which you installed Oracle first.

To configure ASM for the Oracle database

Note: Create disk groups using Oracle ASM Configuration Assistant only on one server in the VCS cluster. These disk groups will be available on the remaining servers in the VCS cluster because it hosts the database of the disk groups.

1. Set the *DISPLAY* variable and start the Oracle ASM Configuration Assistant as Oracle User.

   ```
   # asmca
   ```

2. In the Create ASM Instance dialog box, specify the password for ASM administrator and click **Specify Disk Group** button.
3. In the Create Disk Group dialog box, choose to create a new disk group or add disks to an existing disk group.
4. In the Change Disk Discovery Path dialog box, enter the path name.

   Depending on the disk device you want to use for ASM, enter the disk discovery path.
5. In the Create Disk Group dialog box, do the following:
   - Choose one of the Redundancy options.
   - In the Select Member Disks area, choose the **Show Candidates** option.
   - Select the disks against the **Disk Path** from the list to create the ASM disk group.
   - Click **OK**.

6 Follow the wizard instructions and complete configuring the Oracle ASM.

Note: After you successfully run Oracle ASM Configuration Assistant, run the `$GRID_HOME/bin/crsctl stat res -t -init` command to check if `ora.<DGname>.dg`, a new Oracle resource(s), is created.

7 Copy the PFILE or the SPFILE for ASM instances from the ASM disk group to the local file system.

The VCS agent for ASMInst agent does not support PFILE or SPFILE for ASM instances on the ASM diskgroups.

Enabling the clustering daemon for ASM-managed database

If your database is ASM-managed, then you must make sure that the Oracle clustering daemon is enabled in each node of the cluster. If the CSSD is not enabled, perform the following procedure.

To enable the clustering daemon for ASM-managed database

1 Log in as superuser.

2 Run the following command to configure and start the clustering daemon.

```
# $GRID_HOME/bin/crsctl start has
```

3 Make sure the node has only one Oracle CSSD process running.

Disabling automatic startup of ASM

When the `ocssd.bin` process starts, it automatically starts up the ASM instance. This could cause concurrency violation issues if the ASMInst resource is configured as a part of failover service group in the following scenario. Suppose a node goes down or restarts for some reason, all the resources on this node comes online on the second node. After the reboot, the `ohasd` process automatically starts the `ocssd.bin` process. This automatically starts the ASM instance on node1, which leads to concurrency violation. To resolve this issue, Veritas recommends you to disable automatic startup of ASM.

Run the following command on each node of the cluster:

```
# $GRID_HOME/bin/srvctl disable asm
```

Configuring and starting up ASM on remaining nodes

This procedure is applicable when Oracle grid infrastructure is installed on all nodes and ASM is configured on the first node. See "Configuring Oracle ASM on the first node of the cluster" on page 51.

Configure the remaining nodes by following the procedure given below.

To create ASM and start ASM on remaining nodes

1. Copy the `spfile` for ASM from the ASM Disks that is to be used on the remote node.

   For example:

   ```
   ASMCMD> spget +DATA/asm/asmparameterfile/<registry_file>
   ```

   ```
   ASMCMD> spcopy +DATA/asm/asmparameterfile/<registry_file>
   /u01/oraHome/dbs/spfileASM.ora
   ```

2. Stop the database and the ASM Disk group on the first node.
3. Copy the spfile from the first node to the remote node.
4. Copy *$ORACLE_BASE/admin/SID** from first node to the remote node.
5. Add an asm instance on the remote node.

   For example: `$/u01/product/`*`oracleVersion`*`/grid/bin/srvctl add asm -p /u01/oraHome/dbs/spfileASM.ora`
6. Start the asm instance using the srvctl command.
7. Import the VxVM disk group on the remote node.

To configure database on remaining nodes that use ASM disk-groups

1. Run the `srvctl add database` command to register the Oracle Database(s) that is running on the nodes.

   Note: Use the credentials of the Oracle software owner to register the database. For more information on the complete list of parameters, refer to the Oracle documentation.

2. Login to the Oracle ASM instance running on the remote node.
3. Run the `SQL> ALTER disk group <DGname> mount` command to mount the Oracle ASM disk groups.
4. Repeat Step 3 to mount all the required disk groups.

5 Run the `$GRID_HOME/bin/crsctl stat res -t -init` command to check if the disk groups are auto-registered to OHASD.

The output displays *ora.<DGname>.dg* for the registered disk groups.

6 Run the `$GRID_HOME/bin/srvctl modify database -a <diskgroup_list>` command to add the Oracle ASM disk groups as a dependency to the database(s).

Installing Oracle binaries on the first node of the cluster

Install a version of Oracle that the Cluster Server agent for Oracle supports.

Note: If you want to use Oracle ASM feature, do not disable the Cluster Synchronization Services.

To install the Oracle binaries, follow the instructions in the Oracle documentation.

Disable the Oracle clustering daemon.

See "Disabling the clustering daemon" on page 54.

If you want to use ASM, skip this step and proceed to configuring the Oracle ASM.

See "Configuring Oracle ASM on the first node of the cluster" on page 51.

Proceed to configure the database on shared disks.

See "Configuring the Oracle database" on page 55.

Disabling the clustering daemon

If you installed Oracle binaries on shared disks, you must disable the Oracle clustering daemon.

Note: If you want to use ASM feature, then do not disable the Oracle clustering daemon.

Oracle provides a clustering daemon called Oracle Cluster Synchronization Service Daemon (CSSD). If the Oracle binary files are on a shared storage, the `init` command to start the daemon may result in an error. Because a VCS cluster for Oracle does not require this daemon, Veritas recommends you to disable the daemon.

To disable the daemon on Solaris 11

- On the node from where you ran the Oracle installation wizard:
  - Find out the Fault Management Resource Identifier (FMRI) for the cssd daemon. Type the following from the command prompt:

    ```
    # svcs | grep cssd
    ```

    The FMRI for the cssd daemon is displayed.
  - Modify the Service Configuration Repository such that the SVC does not control the daemon. Type the following from the command prompt:

    ```
    # svccfg delete -f <FMRI>
    ```

    FMRI is the cssd FMRI you obtained.

Configuring the Oracle database

Configure an Oracle database on shared disks using the Database Configuration Assistant. You need to configure the database only once, from the node on which you installed Oracle first.

If you configured ASM for Oracle database storage, configure the Oracle database on ASM disk groups.

To configure the Oracle database

1. Set the *DISPLAY* variable and start the Oracle Database Configuration Assistant as Oracle User.

   ```
   # dbca
   ```

2. Follow the wizard instructions and select other options.
3. In the Storage Options dialog box, select the **File System** or **Raw Devices** option and click **Next**.

   If you configured ASM, select **Automatic Storage Management (ASM)** and click **Next**.
4. If you configured ASM, in the ASM Disk Groups dialog box, select the ASM disk group for the Oracle database storage and click **Next**.
5. In the Database File Locations dialog box, specify a location on the shared disk for the database files to be created and click **Next**.

   If you configured ASM, choose **Use Oracle-Managed Files**, specify the database area, and click **Next**.

6 In the Recovery Configuration dialog box, specify the **Flash Recovery Area** on the shared disk and click **Next**.

If you configured ASM, specify the appropriate **Flash Recovery Area** location and click **Next**.

7 Follow the wizard instructions and complete creating the database.

Copying the $ORACLE_BASE/admin/SID directory

Follow the instruction on each node in the cluster, only if the $ORACLE_HOME is on local disks.

To copy the $ORACLE_BASE/admin/SID directory

- Remote copy $ORACLE_BASE/admin/*SID* on the node where you first configured the database instance to the local disk ($ORACLE_BASE/admin) on other nodes in the cluster.

  The variable *SID* represents the database instance.

  Refer to Oracle's documentation on OFA Structure for more information.

Copying the Oracle ASM initialization parameter file

If your Oracle is ASM-managed, then perform the procedure on each node in the cluster.

To copy the ASM initialization parameter file

- Copy the init*SID*.ora file from the $GRID_HOME/dbs directory on the node where you first installed Oracle ASM on each node of the cluster.

  where *SID* is the ASM instance identifier.

Verifying access to the Oracle database

You can verify access to the database by running an SQL query. Depending on the location of your $ORACLE_HOME, perform one of the following procedures.

Upon completion, the Oracle database is started from the node in the cluster that you last performed this procedure.

To verify access to the Oracle database

- Depending on the location of your $ORACLE_HOME, perform one of the following procedures:

For $ORACLE_HOME on shared disks, start the Oracle database.

Verify that you can access the shared database from each node. Perform the steps from each node that would be a part of the Oracle cluster.

To start the Oracle database (for $ORACLE_HOME on shared disks)

1. If the Oracle binaries are mounted on any other node, unmount it.
2. If the data files on shared disk are mounted on any node, unmount it.
3. Mount the Oracle binaries and data files.
4. Start the database.

For $ORACLE_HOME on local disks, start the Oracle database.

Verify that you can access the database individually on each node that would be a part of the Oracle cluster.

To start the Oracle database (for $ORACLE_HOME on local disks)

1. Make sure you have installed Oracle binaries on the node.
2. If the data files on shared disk are mounted on any node, unmount it.
3. Mount the data files.
4. Start the database.

For ASM, the $ORACLE_HOME is on local disks and the database is configured on ASM disk groups.

Start the ASM-managed Oracle database.

Verify that you can access the database individually on each node that would be a part of the Oracle cluster.

To start the Oracle database (for $ORACLE_HOME on local disks)

1. Make sure you have installed Oracle binaries on the node.
2. If you use VxVM disks as ASM disks, do the following:
   - If the ASM disk groups are mounted on any node, do the following:
     - Unmount the ASM disk groups from the ASM instance.
     - Stop the ASM instance.
     - Stop the VxVM volumes.
     - Deport the VxVM disk groups.
   - Import the VxVM disk groups on this node and start the volumes.
   - Start the ASM instance.
   - Mount the ASM disk groups.
3. If you use CVM volumes for ASM, do the following:
   - Unmount the ASM disk group from the ASM instance on the node where the disk group is mounted.
   - Mount the ASM disk groups on the other node.
4. If you use raw disks as ASM disks, do the following:
   - If the ASM disk groups are mounted on any node, unmount it.
   - If the ASM instance is not running on this node, then start the ASM instance.
   - Mount the ASM disk groups.
5. Start the database.

Chapter 3

Installing and removing the agent for Oracle

This chapter includes the following topics:

- Before you install or upgrade the agent for Oracle
- Installing the VCS agent for Oracle software
- Upgrading the VCS agent for Oracle
- Disabling the Cluster Server agent for Oracle
- Removing the Cluster Server agent for Oracle

Before you install or upgrade the agent for Oracle

Meet the following prerequisites to install or upgrade the Cluster Server agent for Oracle:

- Make sure that VCS is installed in the cluster.
- Verify that the Oracle Server for Solaris and the add-on Listener are installed and configured.
  See Oracle documentation.
  See "About installing Oracle in a VCS environment" on page 36.

Installing the VCS agent for Oracle software

The agent binaries for Cluster Server agent for Oracle are part of VRTSvcsea package. The VRTSvcsea package is already installed if you chose to install all packages or recommended packages during VCS installation. You must manually

install the VRTSvcsea package if you installed minimal packages during VCS installation.

You can install the Cluster Server agent for Oracle from the product disc. You must install the Oracle enterprise agent on all nodes that will host the Oracle service group.

To install the agent on a Solaris SPARC node

1. Log in as superuser.
2. Make sure that the agent is not already installed. Navigate to /opt/VRTSagents/ha/bin and list the contents of the directory.

   If the command returns the VCS agents for Oracle, you can skip this procedure.
3. Create a temporary directory for installation:

   ```
   # mkdir /tmp/install
   ```
4. Insert the disc into a drive connected to your node:
   - If you are running Solaris volume-management software, the software automatically mounts the disc as /cdrom/cdrom0. Type the following command to go to the location:

     ```
     # cd /cdrom/cdrom0
     ```
   - If you are not running Solaris volume-management software, you must mount the disc manually. For example:

     ```
     # mount -F hsfs -o ro /dev/dsk/c0t6d0s2 /cdrom
     ```

     Where, in this example, /dev/dsk/c0t6d0s2 is the default for the CD drive.After the disc is mounted, go to the location where the package files are present:

     ```
     # cd /cdrom
     ```
5. Install the package from the `pkgs` directory:

   ```
   # pkgadd -d VRTSvcsea.pkg
   ```

   The VRTSvcsea package includes the agent binaries for Oracle, Netlsnr, ASMInst, and ASMDG agents. The package also installs the VCS agents for DB2 and Sybase.

To install the Japanese language pack on a Solaris SPARC node

1. After installing the agent, insert the "Language" disc into a drive connected to the system.
2. Insert the disc into a drive connected to your system:
   - If you are running Solaris volume-management software, the software automatically mounts the disc as /cdrom/cdrom0. Type the following command to go to the location:

     ```
     # cd /cdrom/cdrom0
     ```

   - If you are not running Solaris volume-management software, you must mount the disc manually. For example:

     ```
     # mount -F hsfs -o ro /dev/dsk/c0t6d0s2 /cdrom
     ```

     Where, in this example, /dev/dsk/c0t6d0s2 is the default for the CD drive. After the disc is mounted, go to the location where the package files are present:

     ```
     # cd /cdrom
     ```

3. Install the Japanese package:

   ```
   # pkgadd -d VRTSjacse.pkg
   ```

To install the agent on a Solaris x64 node

1. Log in as superuser.
2. Make sure that the agent is not already installed. Navigate to /opt/VRTSagents/ha/bin and list the contents of the directory.

   If the command returns the VCS agents for Oracle, you can skip this procedure.
3. Create a temporary directory for installation:

   ```
   # mkdir /tmp/install
   ```

4. Insert the disc into a drive connected to your node:
   - If you are running Solaris volume-management software, the software automatically mounts the disc as /cdrom/cdrom0. Type the following command to go to the location:

     ```
     # cd /cdrom/cdrom0
     ```

- If you are not running Solaris volume-management software, you must mount the disc manually. For example:

  ```
  # mount -F hsfs -o ro /dev/dsk/c0t6d0s2 /cdrom
  ```

  Where, in this example, /dev/dsk/c0t6d0s2 is the default for the CD drive. After the disc is mounted, go to the location where the package files are present:

  ```
  # cd /cdrom
  ```

5 Install the package:

```
# pkgadd -d VRTSvcsea.pkg
```

The VRTSvcsea package includes the agent binaries for Oracle, Netlsnr, ASMInst, and ASMDG agents. The package also installs the VCS agents for DB2 and Sybase.

Upgrading the VCS agent for Oracle

The agent binaries for Cluster Server agent for Oracle are part of VRTSvcsea package. The VRTSvcsea package is already upgraded if you chose to upgrade to VCS 7.4.2 on a cluster which has a previous version of Cluster Server agent for Oracle installed.

Before you upgrade the agent, make sure you meet the prerequisites to upgrade the Cluster Server agent for Oracle.

See "Before you install or upgrade the agent for Oracle" on page 59.

Warning: The agent package VRTSvcsea includes the VCS agents for Oracle, Sybase, and DB2. So, the procedure to upgrade the VCS agent for Oracle upgrades all the other agents also. Make sure that you have also met the upgrade prerequisites for the other agents that are part of the VRTSvcsea package.

Upgrading VCS agent for Oracle

Perform the following steps in each node of the VCS cluster.

To upgrade VCS agent for Oracle to use with VCS 7.4.2

1. Stop VCS locally.

   ```
   # hastop -local -force
   ```

2. Remove the earlier version of the VCS agent for Oracle.

   - For VCS 5.1 or later:

     ```
     # pkgrm VRTSvcsea
     ```

   - For VCS 5.0MP3 through 5.1 (excluding 5.1):

     ```
     # pkgrm VRTSvcsor
     # pkgrm VRTScsocw
     ```

3. Install the VCS 7.4.2 agent for Oracle.

   See "Installing the VCS agent for Oracle software" on page 59.

4. Make sure that you have copied the OracleTypes.cf file from the /etc/VRTSagents/ha/conf/Oracle directory to /etc/VRTSvcs/conf/config directory.

   Make sure to update the newly copied OracleTypes.cf file with all the type-level changes that you had made to the older OracleTypes.cf file.

   For example, if you had changed the value of the MonitorInterval attribute from the default 60 to 120 seconds, the OracleTypes.cf file gets updated. You must apply these type-level changes manually to the newly copied OracleTypes.cf file.

5. If you use ASM for database storage, then make sure that you have copied the file OracleASMTypes.cf from the directory /etc/VRTSagents/ha/conf/OracleASM to /etc/VRTSvcs/conf/config directory.

   Make sure to update the newly copied OracleASMTypes.cf file with all the type-level changes that you had made to the older OracleASMTypes.cf file.

6. If you had enabled detail monitoring for the Oracle agent in the previous version, set the value of the LevelTwoMonitorFreq attribute to the same value of that of the DetailMonitor attribute.

   Note: The DetailMonitor attribute is deprecated in VCS 6.0 and later.

7. Restart the VCS engine.

   ```
   # hastart
   ```

Upgrading VCS 3.5 agent for Oracle to use with VCS 7.4.2 (Solaris SPARC)

The Veritas Cluster Server agent for Oracle has resource type changes between versions VCS 3.5 and VCS 7.4.2. So, you need to replace resources of type Sqlnet with resources of type Netlsnr.

To upgrade VCS 3.5 agent for Oracle to use with VCS 7.4.2

1. Save the VCS configuration and stop the VCS engine.

```
# haconf -dump -makero
# hastop -all -force
```

2. On all nodes on which the Oracle agent was installed, do the following:
   - Remove the Oracle agent.

```
# pkgrm VRTSvcsor
```

   - Delete the file /etc/VRTSvcs/conf/config/OracleTypes.cf.
   - Install the VCS 7.4.2 agent for Oracle.
     See "Installing the VCS agent for Oracle software" on page 59.

3. Make sure that you have copied the OracleTypes.cf file from the /etc/VRTSagents/ha/conf/Oracle directory to /etc/VRTSvcs/conf/config directory.

   Make sure to update the newly copied OracleTypes.cf file with all the type-level changes that you had made to the older OracleTypes.cf file.

   For example, if you had changed the value of the MonitorInterval attribute from the default 60 to 120 seconds, the OracleTypes.cf file gets updated. You must apply these type-level changes manually to the newly copied OracleTypes.cf file.

4. For Oracle 10g or later, if you use ASM for database storage, then make sure that you have copied the file OracleASMTypes.cf from the directory /etc/VRTSagents/ha/conf/OracleASM to /etc/VRTSvcs/conf/config directory.

5. If you had enabled detail monitoring for the Oracle agent in the previous version, set the value of the LevelTwoMonitorFreq attribute.

6. From a node in the cluster, edit your configuration file /etc/VRTSvcs/conf/config/main.cf.

   Replace all resources of type Sqlnet with resources of type Netlsnr.

   See "About the resource type and attribute definitions" on page 115.

7 Verify the configuration.

```
# hacf -verify config
```

8 Start VCS on local node first.

9 Start VCS on other nodes.

Disabling the Cluster Server agent for Oracle

To disable the Cluster Server agent for Oracle, you must change the Oracle service group to an OFFLINE state. You can stop the application completely or switch the agent to another system.

To disable the agent

1 To remove a system from the service group's SystemList, check if the service group is online:

```
# hagrp -state service_group -sys system_name
```

2 If the service group is online, take it offline. Use one of the following commands:

- To take the service group offline on one node and online it on another node, you can use the -switch option:

```
# hagrp -switch service_group -to system_name
```

- To take the service group offline without bringing it online on any other node in the cluster, enter:

```
# hagrp -offline service_group -sys system_name
```

3 Stop the agent on the node:

```
# haagent -stop Oracle -sys system_name
```

4 When you get the message "`Please look for messages in the log file`," check the file /var/VRTSvcs/log/engine_A.log for a message confirming the agent has stopped.

You can also use the `ps` command to confirm the agent is stopped.

5 You can now remove the service group, the resource type, or both from the VCS configuration after disabling the agent on all nodes.

See the *Cluster Server Administrator's Guide* for more information.

Removing the Cluster Server agent for Oracle

Make sure you disabled the agents on all nodes before you remove the service group, the resource type, or both from the VCS configuration.

Warning: The agent package VRTSvcsea includes the VCS agents for Oracle, Sybase, and DB2. So, the following procedure to remove the VCS agent for Oracle removes all the other agents also.

Perform the procedure to remove the agents on each node in the cluster.

To remove the agent on a node

1. Stop all the resources on all the nodes.

```
# hastop -local
```

2. Remove the VRTSvcsea package. Answer prompts accordingly.

```
# pkgrm VRTSvcsea
```

To remove the Japanese language pack on a Solaris SPARC node

- Remove the Japanese language package.

```
# pkgrm VRTSjacse
```

Chapter 4

Configuring VCS service groups for Oracle

This chapter includes the following topics:

- About configuring a service group for Oracle
- Configuring Oracle instances in VCS
- Before you configure the VCS service group for Oracle
- Configuring the VCS service group for Oracle
- Setting up detail monitoring for VCS agents for Oracle
- Enabling and disabling intelligent resource monitoring for agents manually

About configuring a service group for Oracle

Configuring the Oracle service group involves creating the Oracle 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 a VCS service group for Oracle using one of the following:

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

Note: If you use Oracle ASM feature, you must also configure ASMInst resources and ASMDG resources to keep the ASM instance highly available. You must use the command-line to configure these resources.

Configuring Oracle instances in VCS

You can set up Oracle in different ways in a VCS environment. Configuring Oracle for VCS involves configuring the Oracle files `listener.ora` and `tnsnames.ora` as per VCS requirements.

If you are configuring a service group for an Oracle instance running in a non-global zone, you must make sure that the zone is created on all the systems that will be part of the service group.

Table 4-1 Oracle configurations in VCS

Configuration type	Reference
Single Oracle instance configuration	See "Configuring a single Oracle instance in VCS" on page 68.
Multiple Oracle instances (single listener) configuration	See "Configuring multiple Oracle instances (single listener) in VCS" on page 70.
Multiple Oracle instances (multiple listeners) configuration	See "Configuring multiple Oracle instances (multiple listeners) in VCS" on page 71.
Configuration with shared server support	See "Configuring an Oracle instance with shared server support in VCS" on page 73.

See "Best practices for multiple Oracle instance configurations in a VCS environment" on page 195.

Note: Make sure that the Oracle files `$TNS_ADMIN/listener.ora` and `$TNS_ADMIN/tnsnames.ora` are identical across the cluster nodes. If the files are not identical, remote copy the files from a local node to the other nodes in the cluster.

If a grid infrastructure is set up, make sure that you do not use the default name for the database listener. Otherwise, you may encounter various issues, because the grid infrastructure starts a listener with its default name, which is LISTENER.

Configuring a single Oracle instance in VCS

Review the resource dependency graph and sample configuration of service groups involving a single Oracle instance.

See "Sample single Oracle instance configuration" on page 140.

To configure an Oracle instance in VCS

1. Review the Oracle and Netlsnr resource types and their attribute definitions.

   See "About the resource type and attribute definitions" on page 115.

2. Configure the Oracle file tnsnames.ora as per VCS requirements. The changes required in the file depends on your Oracle configuration.

   For clients to connect to the failover instance, in the file tnsnames.ora located at $TNS_ADMIN, change the host name for all TCP protocol address databases to the virtual IP address for the service group.

   The following example assumes that the host name for the database is set to oraprod, which represents the virtual IP address for the service group.

   ```
   prod =
     (DESCRIPTION =
      (ADDRESS_LIST =
       (ADDRESS = (PROTOCOL = TCP)(HOST = oraprod)(PORT = 1521))
      )
      (CONNECT_DATA =
       (SERVICE_NAME = prod)
      )
     )
   ```

3. Configure the Oracle file listener.ora as per VCS requirements. The changes required in the file depends on your Oracle configuration.

   In the file listener.ora located at $TNS_ADMIN, edit the "Host=" line in the ADDRESS_LIST section and add the name of the high availability address for the service group, in this case, oraprod.

   ```
   LISTENER_PROD =
     (DESCRIPTION_LIST =
     (DESCRIPTION =
      (ADDRESS_LIST =
       (ADDRESS = (PROTOCOL = TCP)(HOST = oraprod)(PORT = 1521))
      )
     )
   ```

4. Create the VCS service groups for Oracle.

   See "Configuring the VCS service group for Oracle" on page 77.

5. Bring the Oracle service group online.

   See "Bringing the service group online" on page 94.

Configuring multiple Oracle instances (single listener) in VCS

Review the resource dependency graph and sample configuration of a service group involving multiple Oracle instances.

See "Sample multiple Oracle instances (single listener) configuration" on page 143.

To configure multiple Oracle instances (single listener)

1. Review the Oracle and Netlsnr resource types and their attribute definitions.

   See "About the resource type and attribute definitions" on page 115.

2. Configure the Oracle file tnsnames.ora as per VCS requirements. The changes required in the file depends on your Oracle configuration.

   For clients to connect to the failover instance, in the file tnsnames.ora located at $TNS_ADMIN, change the host name for all TCP protocol address databases to the virtual IP address for the service group.

   The following example assumes that the host name for the database is set to oraprod, which represents the virtual IP address for the service group.

```
prod =
  (DESCRIPTION =
   (ADDRESS_LIST =
    (ADDRESS = (PROTOCOL = TCP)(HOST = oraprod)(PORT = 1521))
   )
   (CONNECT_DATA =
    (SERVICE_NAME = prod)
   )
  )
 mktg =
  (DESCRIPTION =
   (ADDRESS_LIST =
    (ADDRESS = (PROTOCOL = TCP)(HOST = oramktg)(PORT = 1522))
   )
   (CONNECT_DATA =
    (SERVICE_NAME = mktg)
   )
  )
```

3 Configure the Oracle file listener.ora as per VCS requirements. The changes required in the file depends on your Oracle configuration.

In the file listener.ora located at $TNS_ADMIN, edit the "Host=" line in the ADDRESS_LIST section and add the name of the high availability address for the service group, in this case, oraprod.

```
LISTENER_ORACLE =
 (DESCRIPTION_LIST =
 (DESCRIPTION =
 (ADDRESS_LIST =
  (ADDRESS = (PROTOCOL = TCP)(HOST = oraprod)(PORT = 1521))
  (ADDRESS = (PROTOCOL = TCP)(HOST = oramktg)(PORT = 1522))
 )
```

4 Create the VCS service groups for Oracle.

See "Configuring the VCS service group for Oracle" on page 77.

5 Bring the Oracle service group online.

See "Bringing the service group online" on page 94.

Configuring multiple Oracle instances (multiple listeners) in VCS

Review the resource dependency graph and sample configuration of a service group involving multiple Oracle instance.

See "Sample multiple instance (multiple listeners) configuration" on page 150.

To configure multiple Oracle instances (multiple listeners)

1. Review the Oracle and Netlsnr resource types and their attribute definitions.

   See "About the resource type and attribute definitions" on page 115.

2. Configure the Oracle file tnsnames.ora as per VCS requirements. The changes required in the file depends on your Oracle configuration.

   For clients to connect to the failover instance, in the file tnsnames.ora located at $TNS_ADMIN, change the host name for all TCP protocol address databases to the virtual IP address for the service group.

```
prod =
  (DESCRIPTION =
  (ADDRESS_LIST =
   (ADDRESS = (PROTOCOL = TCP)(HOST = oraprod)(PORT = 1521))
  )
  (CONNECT_DATA =
   (SERVICE_NAME = prod)
  )
 )
 mktg =
  (DESCRIPTION =
  (ADDRESS_LIST =
   (ADDRESS = (PROTOCOL = TCP)(HOST = oramktg)(PORT = 1522))
  )
  (CONNECT_DATA =
   (SERVICE_NAME = mktg)
  )
 )
```

3 Configure the Oracle file listener.ora as per VCS requirements. The changes required in the file depends on your Oracle configuration.

In the file listener.ora, create independent configurations for each listener.

```
LISTENER_PROD =
  (DESCRIPTION_LIST =
  (DESCRIPTION =
  (ADDRESS_LIST =
   (ADDRESS = (PROTOCOL = TCP)(HOST = oraprod)(PORT = 1521))
  )
 )

 LISTENER_MKTG =
  (DESCRIPTION_LIST =
  (DESCRIPTION =
  (ADDRESS_LIST =
   (ADDRESS = (PROTOCOL = TCP)(HOST = oramktg)(PORT = 1522))
  )
 )
```

4 Create the VCS service groups for Oracle.

See "Configuring the VCS service group for Oracle" on page 77.

5 Bring the Oracle service group online.

See "Bringing the service group online" on page 94.

Configuring an Oracle instance with shared server support in VCS

Review the resource dependency graph and sample configuration.

See "Sample Oracle configuration with shared server support" on page 154.

To configure Oracle with shared server support

1. Review the Oracle and Netlsnr resource types and their attribute definitions.

   See "About the resource type and attribute definitions" on page 115.

2. Configure the Oracle file tnsnames.ora as per VCS requirements. The changes required in the file depends on your Oracle configuration.

   For clients to connect to the failover instance, in the file tnsnames.ora located at $TNS_ADMIN, change the host name for all TCP protocol address databases to the virtual IP address for the service group.

   The following example assumes that the host name for the database is set to oraprod, which represents the virtual IP address for the service group.

   ```
   prod =
     (DESCRIPTION =
     (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = oraprod)(PORT = 1521))
     )
     (CONNECT_DATA =
      (SERVICE_NAME = prod)
     )
    )
   ```

3. Configure the Oracle file listener.ora as per VCS requirements. The changes required in the file depends on your Oracle configuration.

   In the file listener.ora located at $TNS_ADMIN, edit the "Host=" line in the ADDRESS_LIST section and add the name of the high availability address for the service group, in this case, oraprod.

   ```
   LISTENER_PROD =
     (DESCRIPTION_LIST =
     (DESCRIPTION =
     (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = oraprod)(PORT = 1522))
     )
    )
   ```

4 In the initialization parameter file, for the dispatchers parameter, set the host name for all TCP protocol address dispatchers to the virtual IP address for the service group.

In the following example, the host name for the dispatcher is set to oraprod.

```
dispatchers =
    "(ADDRESS = (PROTOCOL = TCP)(HOST = oraprod))
    (MUL=ON)(TICK=15)(POOL=true)"
```

5 In the initialization parameter file, set the LOCAL_LISTENER attribute to a listener alias, so that the Oracle PMON process registers information with the listener.

```
local_listener=listener_alias
```

This listener alias gets appended by the default domain set in the file sqlnet.ora.

6 In the file tnsnames.ora, create an entry for the listener alias to resolve the address information. In the address parameter, set the host name to the virtual IP address for the service group, without the CONNECT_DATA portion of the connect descriptor.

In the following example, the listener *listener_alias* uses TCP/IP port 1521 and the virtual IP address for the service group is oraprod.

```
listener_alias=
    (address = (PROTOCOL = TCP)(HOST = oraprod)
    (PORT= 1521))
```

7 Create the VCS service groups for Oracle.

See "Configuring the VCS service group for Oracle" on page 77.

8 Bring the Oracle service group online.

See "Bringing the service group online" on page 94.

Before you configure the VCS service group for Oracle

Before you configure the Oracle service group, you must:

- Verify that VCS is installed and configured on all nodes in the cluster where you will configure the service group.
  Refer to the *Cluster Server Installation Guide* for more information.

- Verify that Oracle is installed and configured identically on all nodes in the cluster.
  See "About installing Oracle in a VCS environment" on page 36.
- Verify that the Cluster Server agent for Oracle is installed on all nodes in the cluster.
  See "Before you install or upgrade the agent for Oracle" on page 59.
- Verify that the type definition for Cluster Server agent for Oracle is imported into the VCS engine.
  See "Importing the type definition files for Cluster Server agent for Oracle" on page 76.
- If the Oracle database is ASM-managed, make sure that the Oracle clustering daemon is enabled. Else, the OCSSD daemon must be disabled.

Importing the type definition files for Cluster Server agent for Oracle

Perform this procedure if you installed the VRTSvcsea package manually. The installer program imports the type definition files if you chose all packages or recommended packages during VCS installation.

Before configuring the Cluster Server agent for Oracle, you must import the OracleTypes.cf file to the VCS engine.

If you want to configure the ASM agents, you must import the OracleASMTypes.cf file. Import the OracleASMTypes.cf file using the command-line.

To import using the Cluster Manager

1. Start Cluster Manager and log on to the cluster.
2. From the Cluster Explorer's **File** menu, choose **Import Types**.
3. In the Import Types dialog box, select the /etc/VRTSagents/ha/conf/Oracle/OracleTypes.cf file.
4. Click **Import**.
5. Save the configuration.

To import using the command line

1. Log in to the system as superuser.
2. Ensure that all changes to the existing configuration have been saved and that further changes are prevented while you modify main.cf:

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

3 To ensure that VCS is not running while you edit main.cf, stop the VCS engine on all nodes and leave the resources available:

```
# hastop -all  -force
```

4 Make a backup copy of the main.cf file:

```
# cd /etc/VRTSvcs/conf/config
# cp main.cf main.cf.orig
```

5 Copy the Oracle types configuration file into place:

```
# cp  /etc/VRTSagents/ha/conf/Oracle/OracleTypes.cf \
/etc/VRTSvcs/conf/config/OracleTypes.cf
```

If you want to configure the ASM agents, do the following:

```
# cp /etc/VRTSagents/ha/conf/OracleASM/OracleASMTypes.cf \
/etc/VRTSvcs/conf/config/OracleASMTypes.cf
```

6 Edit the main.cf file to include the OracleTypes.cf file:

```
include OracleTypes.cf
```

If you want to configure the ASM agents, you must also include the OracleASMTypes.cf file:

```
include "OracleASMTypes.cf"
```

7 Verify the configuration.

```
# cd /etc/VRTSvcs/conf/config
# hacf -verify .
```

8 Start the VCS engine.

```
# hastart
```

Configuring the VCS service group for Oracle

You can configure Oracle in a VCS environment in one of the ways that VCS supports.

See “Configuring Oracle instances in VCS” on page 68.

Table 4-2 lists the interfaces available to configure the VCS agent for Oracle after you have configured Oracle and installed the agent.

Table 4-2 Interfaces to configure VCS agent for Oracle

Configuration method	Reference
The Cluster Manager (Java console)	See "Configuring the VCS service group for Oracle using Cluster Manager (Java console)" on page 78.
The command-line	See "Configuring the VCS service group for Oracle using the command-line" on page 80.
Veritas InfoScale Operations Manager	Refer to the Veritas InfoScale Operations Manager documentation for more details.

Review the following to configure the service group:

- Sample configuration files and resource dependency graphs of the Oracle service group.
  See "About the sample configurations for Oracle enterprise agent" on page 139.
- Resource type and the attribute definitions of the Oracle and Netlsnr agents.
  For ASM, resource type and the attribute definitions of the ASMInst and ASMDG agents.
  See "About the resource type and attribute definitions" on page 115.

Configuring the VCS service group for Oracle using Cluster Manager (Java console)

A template for the Oracle resource groups is automatically installed when you install the Oracle enterprise agent. Using the VCS Cluster Manager, you can view the template, which displays the Oracle service group, its resources and their attributes. You can dynamically modify the attribute values as necessary for your configuration.

Configuration tasks involve:

- Creating a new service group
- Editing the resource attributes

Note: Using the Java Console, you cannot configure a service group for an Oracle instance running in Solaris non-global zones. Use the configuration wizard to configure such service groups.

To create a new service group

1. Make sure that the Oracle type definition file OracleTypes.cf is imported in your configuration.
2. Launch the Cluster Configuration wizard. Do one of the following:
   - From the Cluster Explorer menu, select **Tools > Configuration Wizard**.
   - If no service groups exist on the system, **Cluster Explorer** prompts you to launch the **Cluster Configuration wizard**. Click **Yes** when prompted.

   The Loading Templates Information window appears, and launches the wizard.
3. Read the information on the Welcome screen and click **Next**.
4. Specify the name of the service group and the target systems on which it is to be configured.
   - Enter the name of the service group.
   - On the **Available Systems** box, select the systems on which to configure the service group.
   - Click the right arrow to move the selected systems to the **Systems for Service Group** box. To remove a system from the box, select the system and click the left arrow.
   - Indicate the order in which the systems will fail over (priority). System priority is numbered sequentially, with 1 denoting the system that starts first following a failover.
   - Click **Next**.
5. In the Would you like to use a template to configure the service group? dialog box, click **Next**.
6. In the Select a template to configure the service group dialog box, select the OracleGroup template on which to base the service group.

   If applicable, a window opens notifying you that names of some resources within the new service group are already in use. This window also includes a list of alternative names. You can select an alternative name, or modify the name. Click **Next** after resolving the name clashes.
7. Click **Next** to create the service group based on the selected template.

   A window opens indicating that commands are being sent to add the group, its resources, and the attributes and dependencies specified in the template.

   A progress indicator displays the percentage of the commands fired.

   The actual commands are displayed at the top of the indicator.

8 Click **Next** when prompted that the service group has been successfully created.

A window opens listing the service group's resources and their associated attributes.

9 Click **Finish** to accept the default values and complete the configuration.

You can modify the default values of the resources according to your specific configuration requirements, as instructed in the following section.

To edit the resource attributes

1 Select the resource from the list on the left pane.

The resource's attributes appear in the right pane.

2 Select the attribute to be modified.

See "About the resource type and attribute definitions" on page 115.

3 Click the edit icon in the **Edit** column.

4 In the Edit Attribute dialog box, enter the attribute values. To modify the scope of the attribute, click the option buttons for **Global** or **Local**.

5 Click **OK**.

6 Repeat the procedure for each resource and its attributes and click **Finish**. Edit the attributes for all the resources according to your configuration.

7 Follow the wizard instructions to complete the configuration.

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

Configuring the VCS service group for Oracle using the command-line

The Cluster Server agent for Oracle comes with a sample configuration file that can be used as reference to directly modify your present configuration file. When you use this method, you must stop and restart VCS to implement the configuration.

The Cluster Server agent for Oracle also includes sample configuration files for ASM configurations in a VCS environment.

See "About the sample configurations for Oracle enterprise agent" on page 139.

To configure the service group using the sample configuration file

1 Log in as superuser on the system where oracle is running.

2 Ensure that all changes to the existing configuration have been saved and that further changes are prevented while you modify main.cf:

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

3 To ensure that VCS is not running while you edit main.cf, issue the following command to stop the VCS engine on all systems and leave the resources available:

```
# hastop -all  -force
```

4 Make a backup copy of the main.cf file:

```
# cd /etc/VRTSvcs/conf/config
# cp main.cf main.cf.orig
```

5 Edit the main.cf file.

You can use the file /etc/VRTSagents/ha/conf/Oracle/sample_main.cf for reference.

Do the following:

- Create the Oracle and Netlsnr resources.
  For ASM instances and disk groups, create the ASMInst and ASMDG resources.
- Edit the default attributes to match the parameters in your configuration.
  If you have configured ASMInst agent in a zone environment, then set the value of the resource type attribute OnlineRetryLimit to 1.
  See "About the resource type and attribute definitions" on page 115.
- Assign dependencies to the newly created resources.
  See "About the sample configurations for Oracle enterprise agent" on page 139.
  See the *Cluster Server Administrator's Guide* for more information on assigning dependencies.

6 Save and close the file.

7 Verify the syntax of the file /etc/VRTSvcs/conf/config/main.cf:

```
# cd /etc/VRTSvcs/conf/config/
# hacf -verify .
```

8 Start the VCS engine:

```
# hastart
```

9 Verify that all Oracle service group resources are brought online:

```
# hagrp -state
```

10 Take the service group offline and verify that all resources are stopped:

```
# hagrp -offline service_group -sys system_name
# hagrp -state
```

11 Bring the service group online again and verify that all resources are available:

```
# hagrp -online service_group -sys system_name
# hagrp -state
```

12 Start the VCS engine on all remaining nodes in the cluster:

```
# hastart
```

13 (Optional) Switch the Oracle service group to to one of the cluster node to verify that it has been configured correctly for failover:

```
# hagrp -switch service_group -to any
```

Verify that all Oracle service group resources are brought online on the system:

```
# hagrp -state
```

14 On all the nodes, look at the following log files for any errors or status:

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

/var/VRTSvcs/log/ASMInst_A.log
/var/VRTSvcs/log/ASMDG_A.log
```

Encrypting Oracle database user and listener passwords

VCS provides a utility to encrypt database user passwords and listener passwords. You must encrypt the Pword attribute in the Oracle agent and the LsnrPwd attribute in the Netlsnr agent before you configure these attributes.

See the `vcsencrypt` manual page.

The vcsencrypt utility also allows you to encrypt the agent passwords using a security key. The security key supports AES (Advanced Encryption Standard) encryption which creates a more secure password for the agent.

See the *Cluster Server Administrator's Guide* for more information.

Note: You need not encrypt passwords if you use the VCS Cluster Manager (Java Console) to configure attributes.

The user passwords that are used for detail monitoring of the Oracle database are encrypted. The listener password that is considered for querying the status of the listener and stopping the listener is also encrypted.

Oracle provides the option of storing the listener password in the listener.ora file, in both clear text and encrypted formats. Irrespective of the format in which the password is stored in Oracle, you must encrypt the password using the vcsencrypt utility before you configure the LsnrPwd attribute.

If you encrypted the listener password using the Oracle lsnrctl utility, do the following depending on the Oracle version you use:

- Pass the clear text password to the vcsencrypt utility.
  For example, if the listener password is “oracle” pass this clear text password to the vcsencrypt utility as follows:

  ```
  # vcsencrypt -agent oracle
  ```

  The vcsencrypt utility displays the encrypted password. For example:
  bnqPdnCneNnnG

The `vcsencrypt` utility also allows you to encrypt the agent passwords using a security key. The security key supports AES (Advanced Encryption Standard) encryption which creates a secure password for the agent. See the *Cluster Server Administrator’s Guide* for more information.

Setting up detail monitoring for VCS agents for Oracle

The Cluster Server 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.

  Note: The agent for Oracle ASM does only basic monitoring for ASM instances.

- 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 in traditional database and container database" on page 21.

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 Cluster Server 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. Veritas 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 VCS configuration writable:

   ```
   # haconf -makerw
   ```

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

   ```
   # hagrp -freeze service_group
   ```

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.

For traditional or container databases:

```
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);
```

Note: For a CDB resource, the user name should be prefixed with C##.

To enable detail monitoring for PDB, issue the following statements at the sqlplus prompt to create the test table:

```
$ORACLE_HOME/bin/sqlplus /nolog
connect / as sysdba
```

Create common user for CDB if not already created.

```
create user <User>
identified by <Pword>
default tablespace USERS
temporary tablespace TEMP
quota 100K on USERS;
grant connect to <User> container=all;
alter session set container=PDBName
create table <User>.<Table> ( tstamp date );
insert into <User>.<Table> (tstamp) values (SYSDATE);
```

Note: The VCS Agent for Oracle requires a common user for both CDB and PDB detail monitoring, else detail monitoring will not work for PDB.

7 To test the database table for use, do the following for CDB or traditional databases:

```
$ORACLE_HOME/bin/sqlplus /nolog
connect / as sysdba
connect  <User>/<Pword>
update <User>.<Table> set(tstamp)=SYSDATE;
select TO_CHAR(tstamp, 'MON DD, YYYY HH:MI:SS AM')
from <User>.<Table>;
```

For PDB

```
alter session set container=PDBName;
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 VCS commands:

```
# hares -modify OracleResource User <User>
# hares -modify OracleResource Pword <Pword>
# hares -modify OracleResource Table <Table>
# hares -modify OracleResource MonScript "./bin/Oracle/SqlTest.pl"

# hares -override OracleResource LevelTwoMonitorFreq
# hares -modify OracleResource LevelTwoMonitorFreq 1

# haconf -dump  -makero

# hagrp -unfreeze service_group
```

You can also use Cluster Manager (Java Console) to set these attributes.

Enabling and disabling detail monitoring for Oracle resource

Review the instructions to enable or disable detail monitoring.

To enable detail monitoring

- Set the value of the LevelTwoMonitorFreq attribute.

```
# hares -override OracleResource LevelTwoMonitorFreq
# hares -modify OracleResource LevelTwoMonitorFreq 1
```

To disable detail monitoring

- Set the value of the LevelTwoMonitorFreq attribute to 0.

```
# hares -modify OracleResource LevelTwoMonitorFreq 0
```

Setting up detail monitoring for Netlsnr

For Netlsnr agent, the detail monitoring is not enabled by default to monitor the listener process.

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

To disable detail monitoring for Netlsnr

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

```
# haconf -makerw
# hagrp -freeze service_group
# hares -modify LsnrResource MonScript ""

# hares -override LsnrResource LevelTwoMonitorFreq
# hares -modify LsnrResource LevelTwoMonitorFreq 0

# haconf -dump -makero
# hagrp -unfreeze
```

To set up detail monitoring for Netlsnr

1 Make the VCS configuration writable:

```
# haconf -makerw
```

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

```
# hagrp -freeze service_group
```

3 Enable detail monitoring by entering the following commands:

```
# hares -modify LsnrResource MonScript "./bin/Netlsnr/LsnrTest.pl"

# hares -override LsnrResource LevelTwoMonitorFreq
# hares -modify LsnrResource LevelTwoMonitorFreq 1

# haconf -dump -makero

# hagrp -unfreeze service_group
```

Enabling and disabling intelligent resource monitoring for agents manually

Review the following procedures to enable or disable intelligent resource monitoring manually. The intelligent resource monitoring feature is enabled by default. The IMF resource type attribute determines whether an IMF-aware agent must perform intelligent resource monitoring.

See "How the Oracle and Netlsnr agents support intelligent resource monitoring" on page 11.

To enable intelligent resource monitoring

1 Make the VCS configuration writable.

```
# haconf -makerw
```

2 Run the following command to enable intelligent resource monitoring.

- To enable intelligent monitoring of offline resources:

  ```
  # hatype -modify resource_type IMF -update Mode 1
  ```

- To enable intelligent monitoring of online resources:

```
# hatype -modify resource_type IMF -update Mode 2
```

- To enable intelligent monitoring of both online and offline resources:

```
# hatype -modify resource_type IMF -update Mode 3
```

3 If required, change the values of the MonitorFreq key and the RegisterRetryLimit key of the IMF attribute.

See "Attribute definitions for the Oracle agent" on page 116.

See "Attribute definition for the Netlsnr agent" on page 127.

4 Save the VCS configuration.

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

5 Make sure that the AMF kernel driver is configured on all nodes in the cluster.

```
/lib/svc/method/amf status
```

If the AMF kernel driver is configured, the output resembles:

```
AMF: Module loaded and configured
```

Configure the AMF driver if the command output returns that the AMF driver is not loaded or not configured.

See "Administering the AMF kernel driver" on page 91.

6 Restart the agent. Run the following commands on each node.

```
# haagent -stop agent_name -force -sys sys_name
# haagent -start agent_name -sys sys_name
```

To disable intelligent resource monitoring

1 Make the VCS configuration writable.

```
# haconf -makerw
```

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

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

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

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

4 Save the VCS configuration.

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

Note: VCS provides haimfconfig script to enable or disable the IMF functionality for agents. You can use the script with VCS in running or stopped state. Use the script to enable or disable IMF for the IMF-aware bundled agents, enterprise agents, and custom agents.

Administering the AMF kernel driver

Review the following procedures to start, stop, or unload the AMF kernel driver.

To start the AMF kernel driver

1 Set the value of the AMF_START variable to 1 in the following file, if the value is not already 1:

```
# /etc/default/amf
```

2 Start the AMF kernel driver. Run the following command:

```
# svcadm enable amf
```

To stop the AMF kernel driver

1 Set the value of the AMF_STOP variable to 1 in the following file, if the value is not already 1:

```
# /etc/default/amf
```

2 Stop the AMF kernel driver. Run the following command:

```
# svcadm disable amf
```

To unload the AMF kernel driver

1 If agent downtime is not a concern, use the following steps to unload the AMF kernel driver:

- Stop the agents that are registered with the AMF kernel driver.
  The `amfstat` command output lists the agents that are registered with AMF under the Registered Reapers section.
  See the `amfstat` manual page.
- On Solaris 11, set the value of the AMF_DISABLE variable to 1 in the following file (before stopping the AMF kernel driver):

  ```
  # /etc/default/amf
  ```

  Note: Reset the AMF_DISABLE variable in the /etc/default/amf file before starting AMF kernel driver.

- Stop the AMF kernel driver.
  See "To stop the AMF kernel driver" on page 91.
- Start the agents.

2 If you want minimum downtime of the agents, use the following steps to unload the AMF kernel driver:

- Run the following command to disable the AMF driver even if agents are still registered with it.

  ```
  # amfconfig -Uof
  ```

- On Solaris 11, set the value of the AMF_DISABLE variable to 1 in the following file (before stopping the AMF kernel driver):

  ```
  # /etc/default/amf
  ```

  Note: Reset the AMF_DISABLE variable in the /etc/default/amf file before starting AMF kernel driver.

- Stop the AMF kernel driver.
  See "To stop the AMF kernel driver" on page 91.

Chapter 5

Administering VCS service groups for Oracle

This chapter includes the following topics:

- About administering VCS service groups
- Bringing the service group online
- Taking the service group offline
- Switching the service group
- Modifying the service group configuration

About administering VCS service groups

You can administer service groups in Cluster Server using one of the following interfaces:

- Veritas InfoScale Operations Manager
  See the Veritas InfoScale Operations Manager documentation for more information.
- The Cluster Manager (Java Console)
  See “Bringing the service group online” on page 94.
  See “Taking the service group offline” on page 94.
  See “Switching the service group” on page 94.
  See “Modifying the service group configuration” on page 95.
- The command-line
  See the *Cluster Server Administrator’s Guide* for more information.

Bringing the service group online

Perform the following steps to bring the service group online.

To bring a service group online

1. In the Cluster Explorer configuration tree, select the newly created service group.
2. Right-click the service group name, and select **Enable Resources** to enable all resources in the service group.
3. Right-click the service group name, and select the systems on which to enable the service group (Right-click > Enable >*system_name* or Right-click > Enable > All).
4. Save your configuration (File > Close Configuration).
5. Right-click the service group and select to online the service group on the system (Right-click > Online >*system_name*).

Taking the service group offline

Perform the following steps to take the service group offline.

To take a service group offline

1. In the **Service Groups** tab of the Cluster Explorer configuration tree, right-click the service group.

   or

   Select the cluster in the Cluster Explorer configuration tree, select the **Service Groups** tab, and right-click the service group icon in the view panel.
2. Choose **Offline**, and choose the appropriate system from the pop-up menu (Right-click > Offline > *system_name*).

Switching the service group

The process of switching a service group involves taking it offline on its current system and bringing it online on another system.

To switch a service group

1. In the **Service Groups** tab of the Cluster Explorer configuration tree, right-click the service group.

   or

   Select the cluster in the Cluster Explorer configuration tree, select the **Service Groups** tab, and right-click the service group icon in the view panel.

2. Choose **Switch To**, and choose the appropriate system from the pop-up menu (Right-click > Switch To > *system_name*).

Modifying the service group configuration

You can dynamically modify the Cluster Server agent for Oracle agent using one of the supported interfaces to administer VCS.

Chapter 6

Pluggable database (PDB) migration

This chapter includes the following topics:

- Migrating Pluggable Databases (PDB) between Container Databases (CDB)

Migrating Pluggable Databases (PDB) between Container Databases (CDB)

You can use the `hapdbmigrate` utility to perform a planned end-to-end migration of pluggable databases across containers of the same database version. The destination container database may be on the same node or on another node. The PDB must have its own independent storage.

If the PDB has parent groups, they will be unlinked and frozen by the utility during the migration. After successful migration, they will be relinked and unfrozen by the utility.

The configuration must meet the following requirements:

- The PDB and CDB databases must be on Veritas Cluster File System (CFS) or Veritas File system (VxFS).
- There must be no parent resources for the PDB resource.
- The PDB must be plugged in to the source CDB. The PDB resources may or may not be offline.
- The source and destination CDB resources for the migration must be different.
- The version of the destination CDB must be the same as the source CDB.
- The PDB to be migrated is mounted on an independent file system (separate mount points for CDB and PDB datafiles).

- The CDB and PDB resources must be configured in the same service group.

The utility performs the following actions during the migration:

- Unlinks and freezes the parent groups, if any, depending on the source CDB group where the PDB resource to be migrated is configured.
- Takes PDB resources offline.
- Unplugs the PDB from the CDB and creates an XML file, `<pdb_res_name>_<dest_cdb_res_name>.xml` in the XML directory provided by the user.
- Drops the PDB from the CDB keeping the datafiles.
- Takes offline all the PDB child resources. Unlinks the PDB resource from the source CDB resource and deletes the PDB resource. Unlinks and deletes all PDB child resources.
- Recreates the PDB resource and all its children with original dependencies in the destination CDB service group.
- Brings online the PDB child resources on all the nodes where the destination CDB service group is online
- Plugs the PDB in the destination CDB.
- Brings the PDB resource online after successful plugging. Unfreezes and links the parent group to the source CDB group.

The `hapdbmigrate` utility performs certain pre-requisite checks before the migration to verify that the cluster is ready for PDB migration. If the utility encounters any issues, you will need to manually fix the issues.

The utility is present in the `$VCSHOME/bin` directory.

The log files of the migration are located at `$VCSLOG/log/hapdbmigrate.log`. The logs are rotated after the file exceeds 5 MB and is saved in .gz format `hapdbmigrate.log[1..7].gz`.

Note: Ensure that only one instance of the `hapdbmigrate` utility is running at a time.

To migrate Pluggable Databases (PDB) between Container Databases (CDB)

1. Back up the VCS configuration file `/etc/VRTSvcs/conf/config/main.cf`:

   ```
   # cp /etc/VRTSvcs/conf/config/main.cf \
   /etc/VRTSvcs/conf/config/main.cf.save
   ```

2. Verify that the high availability daemon (had) is running on all the nodes in the cluster.

   ```
   # hastatus -summary
   ```

3. Verify that there are no resources in faulted or unknown state.

   ```
   # hares -state|grep FAULTED
   # hares -state|grep UNKNOWN
   ```

4. Verify that the `PDBName` attribute is present for the PDB resource with the correct value in the `main.cf` configuration file.

5. Verify that the source and destination CDB resources are online.

   ```
   # hares -state resname
   ```

6. On the destination CDB, verify the following:

   - The destination CDB is not in `suspended` mode.
     Any instance of the destination CDB is not in mounted state.
     See the Oracle documentation for more information.
   - If any instance of the destination CDB is in `restricted` state, ensure that the PDB resource you want to migrate has the `StartUpOpt` attribute set to `restricted`.

     ```
     # haconf -makerw
     # hares -modify pdb1 StartUpOpt \
     RESTRICTED
     # haconf -dump -makero
     ```

7. Verify that existing dependencies do not conflict with the migration process.

   The PDB child resources must not be dependent on the CDB resource or any of its child resources.

   The PDB child resources must not have any parent, which is not a part of the PDB child hierarchy.

8 Verify that the XML data directory has read and write permissions for the "oracle" user. The XML data directory must be located either on PDB mounts or at a location accessible to both source and destination CDBs.

9 Run the `hapdbmigrate` utility as the root user:

Note: If there are parent groups dependent on the source CDB group, specify the `-ignoreparentgrp` option.

```
# $VCS_HOME/bin/hapdbmigrate -pdbres pdb_resname -cdbres cdb_resname \
-XMLdirectory xml_directory [-ignoreparentgrp] [-prechecks]
  -pdbres: Name of the PDB resource, which needs to be migrated
 -cdbres: Name of the CDB resource, where the PDB needs to migrate
 -XMLdirectory: XML directory location for the unplugged PDB
-ignoreparentgrp: Utility proceeds even
if the PDB group has parent groups
-prechecks: Performs prechecks and validation
-help|h: Prints usage
```

The migration log file is located at `$VCSLOG/log/hapdbmigrate.log`.

10 Verify that the PDB resource is online on the destination CDB.

```
# hares -state pdb_resname
```

11 Relink the parent service group of the source CDB group manually to the destination CDB group, if it depends on the migrated PDB.

```
# haconf -makerw
# hagrp -dep parent_sg
#Parent           Child             Relationship
parent_sg       source_CDB         online local firm
# hagrp -offline parent_sg -any
# hagrp -unlink parent_sg source_CDB
# hagrp -link parent_sg dest_CDB online local firm
# hagrp -dep parent_sg
#Parent           Child             Relationship
parent_sg       dest_CDB           online local firm
# haconf -dump -makero
```

See "Sample configuration of migratable Oracle pluggable database (PDB) resource in main.cf" on page 185.

See "Resource type definition for the Oracle agent" on page 115.

See “Resource type definition for the Netlsnr agent” on page 126.

Chapter 7

Troubleshooting Cluster Server agent for Oracle

This chapter includes the following topics:

- About troubleshooting Cluster Server 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
- Error messages specific to the ASMInst agent
- Error messages specific to the ASMDG agent
- Troubleshooting issues specific to Oracle in a VCS environment
- Verifying the Oracle health check binaries and intentional offline for an instance of Oracle
- Disabling IMF for a PDB resource

About troubleshooting Cluster Server 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 log file, you must access:

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

/var/VRTSvcs/log/ASMInst_A.log
/var/VRTSvcs/log/ASMDG_A.log
```

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

Review the description of the error messages for the following agents and the possible solutions:

- Oracle agent
- Netlsnr agent
- ASMInst agent
- ASMDG agent

Error messages common to the Oracle and Netlsnr agents

Table 7-1 lists the Cluster Server agent for Oracle error messages with the description and a recommended solution, if available.

Table 7-1 Cluster Server 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 <*Oracle_Home*> 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.

Table 7-1 Cluster Server agent for Oracle error messages *(continued)*

Message	Description and solution
File <*env_file*> 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.
Unable to open pipe from %s	Internal error. Solution: Contact Technical Support for further assistance.
Process <*Process_name*> 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 <*monscript*> returned <*output*>	MonScript failed to execute correctly. Solution: Debug MonScript to assess the exact cause of failure.
Monitor procedure <*monscript*> did not exit	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 <*owner_name*> 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 <*monscript*> 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.

Table 7-1 Cluster Server agent for Oracle error messages *(continued)*

Message	Description and solution
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 7-2 lists the error messages for the VCS agent for Oracle with the description and a recommended solution, if available.

Table 7-2 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 <*Oracle_Home*>/bin	The client utility sqlplus 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 <*Oracle_Home*>/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.

Table 7-2 Oracle agent error messages *(continued)*

Message	Description and solution
Oracle <*SID*> 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 <*SID*> 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 (<*SID*>) kill TERM <*PID1, PID2, ...*>	Warning message to indicate that the Oracle processes would be signaled with SIGTERM. Solution: No action required.
Oracle (<*SID*>) kill KILL <*PID1, PID2, ...*>	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.

Table 7-2 Oracle agent error messages *(continued)*

Message	Description and solution
Database in SUSPENDED state	Warning message to indicate that database is in SUSPENDED state.
Resource <*Res_name*> - 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.
VCS ERROR Resource <*Res_name*>: online procedure did not complete within the expected time.	Solution: Increase the Oracle agent's OnlineTimeout interval.

Table 7-2 Oracle agent error messages *(continued)*

Message	Description and solution
Custom script /opt/VRTSagents/ha/bin/Oracle/start_custom_<*SID*>.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_<*SID*>.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.
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 7-2 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!!	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.

Error messages specific to the Netlsnr agent

Table 7-3 lists the Netlsnr agent error messages with the description and a recommended solution, if available.

Table 7-3 Netlsnr 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 <*Listener*> 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 <*Listener*> kill TERM <*PID1, PID2, ...*>	Warning message to indicate that the Listener process would be signaled with SIGTERM. Solution: No action required.
Listener <*Listener*> kill KILL <*PID1, PID2, ...*>	Warning message to indicate that the Listener process would be signaled with SIGKILL. Solution: No action required.

Table 7-3 Netlsnr agent error messages *(continued)*

Message	Description and solution
lsnrctl not found in <Oralce_Home>/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.

Error messages specific to the ASMInst agent

Table 7-4 lists the ASMInst agent error messages with the description and a recommended solution, if available.

Table 7-4 ASMInst agent error messages

Message	Description and solution
Attribute Home cannot have multiple tokens.	The Home attribute of the ASMInst agent has multiple tokens. Solution: Correct the value of the Home attribute.
Cluster Synchronization Service process is not running.	The Oracle CSSD process is not running. Solution: Enable the CSSD process. See "Enabling the clustering daemon for ASM-managed database" on page 52.
Cluster Synchronization Service died or is restarted.	The Oracle CSSD process has died. Solution: Enable the CSSD process. See "Enabling the clustering daemon for ASM-managed database" on page 52.

Error messages specific to the ASMDG agent

Table 7-5 lists the ASMInst agent error messages with the description and a recommended solution, if available.

Table 7-5 ASMDG agent error messages

Message	Description and solution
No ASM Diskgroup name specified, or is null.	The DiskGroups attribute value for ASMDG agent is not specified. Solution: Specify the value of the Diskgroup attribute.
Agent unable to identify state of the ASMDG resource. The asm_diskstring parameter is not set correctly in the ASM initialization parameter file.	The ASM initialization parameter file does not have the correct value for ASM instance to search the ASM disk groups. Solution: Specify the correct value for the asm_diskstring parameter in the ASM initialization parameter file.
Agent unable to identify state of the resource.	The ASMDG agent cannot identify the state of the resource. Solution: Contact Technical Support for further assistance.

Troubleshooting issues specific to Oracle in a VCS environment

Table 7-6 lists any Oracle issues that you may encounter in a VCS environment.

Table 7-6 Oracle common issue

Message	Description and solution
On Solaris x64: The Oracle database instance terminates at regular short intervals.	Solution: The recommendation from Oracle is to set the values of the database initialization parameters db_cache_size and java_pool_size to optimal values.
ORA-15097: Cannot SHUTDOWN ASM instance with connected RDBMS instance.	This message appears for ASM-managed database if you try to offline the Oracle service group within 2-3 minutes after you brought the service group online. Refer to Oracle bug 5045309 for more information.

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 7-7 lists the checks you can verify with Oracle health check binaries.

Table 7-7 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.

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.

Table 7-7 Oracle health check options *(continued)*

Verify	Solution

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_FullVersion>`
	where *<Arch>_<Oracle_FullVersion>* is the system architecture and the Oracle full version.
	For example, `# ./oraapi_64_121010`, where oraapi_64_121010 is a binary built for Oracle version 12 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.

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_FullVersion>`
	where *<Arch>_<Oracle_FullVersion>* is the system architecture and the Oracle full version.
	For example, `# ./oraapi_64_121010`, where oraapi_64_121010 is a binary built for Oracle version 12 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.

Table 7-7 Oracle health check options *(continued)*

Verify	Solution

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_64_121010`
	`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.

Disabling IMF for a PDB resource

To disable IMF for PDB resource perform the following steps:

1 Override the IMF attribute for a particular PDB resource:

```
# hares -override <pdb_resname> IMF
```

2 Verify if the IMF attribute is overridden using the following command:

```
# hares -value pdb1 IMF
Mode    3       MonitorFreq     5       RegisterRetryLimit      3
```

3 Modify the Mode value to 0

```
# hares -modify pdb1 IMF -update Mode 0
```

4 Verify the Mode value using the following command:

```
# hares -value pdb1 IMF
Mode    0       MonitorFreq     5       RegisterRetryLimit      3
```

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
- Resource type definition for the ASMInst agent
- Resource type definition for the ASMDG agent

About the resource type and attribute definitions

The resource type represents the VCS configuration definition of the agent and specifies how the agent is defined in the configuration file main.cf. The Attribute Definitions explain the attributes associated with the agent. The Required attributes explain the attributes that must be configured for the agent to function properly.

Resource type definition for the Oracle agent

The Oracle agent of the Cluster Server agent for Oracle is represented by the Oracle resource type in VCS.

```
type Oracle (
 static str AgentDirectory = "/opt/VRTSagents/ha/bin/Oracle"
 static keylist SupportedActions = { VRTS_GetInstanceName,
      VRTS_GetRunningServices, DBRestrict, DBUndoRestrict,
      DBResume, DBSuspend, DBTbspBackup, PDBMigrate_PreCheck,
      PlugPDB, "home.vfd", "owner.vfd", "getid", "pfile.vfd" }
 static str ArgList[] = { Sid, Owner, Home, Pfile, StartUpOpt,
```

```
    ShutDownOpt, DBAUser, DBAPword,  EnvFile, AutoEndBkup,
    User, Pword, Table, MonScript, Encoding, MonitorOption,
    DBName, ManagedBy, PDBName }
 static int IMF{} = { Mode=3, MonitorFreq=5, RegisterRetryLimit=3 }
 static str IMFRegList[] = { Home, Owner, Sid, MonitorOption }
 str Sid
 str Owner
 str Home
 str Pfile
 str StartUpOpt = STARTUP_FORCE
 str ShutDownOpt = IMMEDIATE
 str DBName
 str ManagedBy = "ADMIN"
 str DBAUser
 str DBAPword
 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
 static boolean AEPTimeout = 1
 static int OnlineWaitLimit = 5
 str PDBName
)
```

Attribute definitions for the Oracle agent

The Oracle agent attributes are classified as required, optional, and internal.

You must assign values to the following 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. For a policy managed database, the Sid attribute should be set to Sid prefix. See "About the Sid attribute in a policy managed database" on page 125.
Owner	string-scalar	The Oracle user who has privileges to start or stop the database instance. 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. **Note:** Do not append a slash (/) at the end of the path.

You can configure the following optional attributes if they are necessary for your configuration.

Table A-2 Optional attributes for Oracle agent

Optional Attributes	Type and Dimension	Definition
ContainerOpts	static-assoc-int	This resource-type level attribute specifies the container options for the Oracle instances that run in the context of Solaris containers (zones or projects). The values for the following keys are only effective when you set the ContainerInfo service group attribute. Veritas recommends that you do not change the following values. ■ RunInContainer (RIC) 1—The Oracle agent monitors the Oracle instances that run inside of the local container. 0—The Oracle agent monitors the Oracle instances that run outside the local container (in the global environment). Default is 1. ■ PassCInfo (PCI) The Oracle agent receives the container information that is defined in the service group's ContainerInfo attribute. Default is 0. See the *Cluster Server Administrator's Guide* and the *Veritas InfoScale 7.4.2 Virtualization Guide*.

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

Optional Attributes	Type and Dimension	Definition
DBAUser	string-scalar	The database user who has sysdba privileges to start or stop the database.
DBAPword	string-scalar	Encrypted password for DBAUser. Encrypt passwords only when entering them using the command-line. Passwords must be encrypted using the VCS Encrypt Utility (`/opt/VRTSvcs/bin/vcsencrypt`). See "Encrypting Oracle database user and listener passwords" on page 82.
StartUpOpt	string-scalar	Startup options for the Oracle instance. This attribute can take the following values for traditional database and container database: ■ STARTUP ■ STARTUP_FORCE ■ RESTRICTED ■ RECOVERDB ■ SRVCTLSTART ■ CUSTOM ■ SRVCTLSTART_FORCE Default is STARTUP_FORCE. See "Startup and shutdown options for the Oracle agent" on page 15. Startup options for the Oracle instance in pluggable database. This attribute can take the following values: ■ STARTUP ■ STARTUP_FORCE ■ RESTRICTED ■ OPEN_RO See "Startup and shutdown options for the pluggable database (PDB)" on page 18.

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 for traditional database and container database: ■ IMMEDIATE ■ TRANSACTIONAL ■ SRVCTLSTOP ■ CUSTOM ■ SRVCTLSTOP_TRANSACT ■ SRVCTLSTOP_ABORT ■ SRVCTLSTOP_IMMEDIATE Default is IMMEDIATE. See "Startup and shutdown options for the Oracle agent" on page 15. Shut down options for the Oracle instance in pluggable database is IMMEDIATE. See "Startup and shutdown options for the pluggable database (PDB)" on page 18.
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. **Note:** This attribute is applicable only for traditional and container databases. See "Using the SPFILE in a VCS cluster" on page 198.

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

Optional Attributes	Type and Dimension	Definition
AutoEndBkup	boolean-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 **Note:** If a node fails during a hot backup of container database or pluggable database, you must set AutoEndBkup attribute of the corresponding CDB resource to 1. When the AutoEndBkup is set to 1 for the CDB, it also ends the backup of both CDB and PDB during Online. See "About VCS requirements for installing Oracle" on page 37. See "Failing over Oracle after a VCS node failure during hot backup" on page 43.
MonitorOption	integer-scalar	Monitor options for the Oracle instance. This attribute can take values 0 or 1. For traditional and container databases: ■ 0—Process check monitoring (recommended) ■ 1—Health check monitoring For pluggable databases: ■ 0—Basic monitoring You must set the value of this attribute as 1 to use the intentional offline functionality of the agent. Default = 0 The agent supports intelligent resource monitoring only for traditional and CDBs when the attribute value is set to 0. See "Monitor options for the Oracle agent in traditional database and container database" on page 21. See "Monitor for the pluggable database" on page 22.

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

Optional Attributes	Type and Dimension	Definition
IMF	integer-association	

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

Optional Attributes	Type and Dimension	Definition
		This resource-type level attribute determines whether the Oracle agent must perform intelligent resource monitoring. You can also override the value of this attribute at resource-level. This attribute includes the following keys: ■ Mode: Define this attribute to enable or disable intelligent resource monitoring. Valid values are as follows: ■ 0—Does not perform intelligent resource monitoring ■ 1—Performs intelligent resource monitoring for offline resources and performs poll-based monitoring for online resources ■ 2—Performs intelligent resource monitoring for online resources and performs poll-based monitoring for offline resources ■ 3—Performs intelligent resource monitoring for both online and for offline resources Default: 3 ■ MonitorFreq: This key value specifies the frequency at which the agent invokes the monitor agent function. The value of this key is an integer. Default: 5 You can set this key to a non-zero value for cases where the agent requires to perform both poll-based and intelligent resource monitoring. If the value is 0, the agent does not perform poll-based process check monitoring. After the resource registers with the AMF kernel driver, the agent calls the monitor agent function as follows: ■ After every (MonitorFreq x MonitorInterval) number of seconds for online resources ■ After every (MonitorFreq x OfflineMonitorInterval) number of seconds for offline resources ■ RegisterRetryLimit: If you enable intelligent resource monitoring, the agent invokes the oracle_imf_register agent function to register the resource with the AMF kernel driver. The value of the RegisterRetryLimit key determines the number of times the agent must retry registration for a resource. If the agent cannot register the resource within the limit that is specified, then intelligent monitoring is disabled until the resource state changes or the value of the Mode key changes.

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

Optional Attributes	Type and Dimension	Definition
		Default: 3 **Note:** IMF is supported only in traditional and container databases. See "Enabling and disabling intelligent resource monitoring for agents manually" on page 89.
MonScript	string-scalar	Pathname to the script provided for detail monitoring. The default (basic monitoring) is to monitor the database PIDs only. **Note:** Second-level monitoring is disabled if the value of the attribute MonScript is invalid or is set to an empty string. The pathname to 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.
User	string-scalar	Internal database user. Connects to the database for detail monitoring.
LevelTwoMonitorFreq	integer-scalar	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 will monitor Oracle in detail. For example, the value 5 indicates that the agent will monitor Oracle in detail every five online monitor intervals. If you manually upgraded to the VCS 7.4.2 agent, and if you had enabled detail monitoring in the previous version, set the value of the LevelTwoMonitorFreq attribute appropriately. **Note:** If you set the AutoEndBkup attribute value to 0, then make sure that the LevelTwoMonitorFreq attribute value is 1 for detail monitoring. Default = 0

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

Optional Attributes	Type and Dimension	Definition
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 VCS Encrypt Utility (`/opt/VRTSvcs/bin/vcsencrypt`). **Note:** This attribute is not applicable for PDB resources. See "Encrypting Oracle database user and listener passwords" on page 82.
Table	string-scalar	Table for update by `User/Pword`.
Encoding	string-scalar	Specifies operating system encoding that corresponds to Oracle encoding for the displayed Oracle output. For example, if Oracle output is in "JAPANESE_JAPAN.JA16EUC," then "eucJP" is the Solaris value for Encoding. Refer to the Oracle and Solaris documentation for respective encoding values. Default is "". **Note:** This attribute is not applicable for PDB resources.
IntentionalOffline		This resource-type level attribute defines how VCS reacts when Oracle is intentionally stopped outside of VCS control. If you stop Oracle traditional and container database 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 in the CDB. **Note:** If a CDB resource is configured with IntentionalOffline, and the configured CDB is brought down outside the VCS control, then the PDB resources will also report as offline. To enable this functionality, set the IntentionalOffline attribute to 1 for the PDB resource. See the *Cluster Server Administrator's Guide*.

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

Optional Attributes	Type and Dimension	Definition
DBName	string-scalar	Set this attribute only when the database is a policy managed RAC database. The value of this attribute must be set to the database unique name.
ManagedBy	string-scalar	Default value for this attribute is ADMIN. In a policy managed RAC database this attribute must be set to POLICY.
PDBName	string-scalar	This attribute must be configured for a PDB. And the value should be set for a PDB database name. For traditional and CDB database, do not set this attribute.

Table A-3 lists the internal attribute for Oracle agent. This attribute is for internal use only. Veritas 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.

About the Sid attribute in a policy managed database

The SID attribute is a required attribute. This section provides information to define the SID attribute in a policy managed database.

The SID prefix comprises of the first 8 alphanumeric characters of the database unique name. It can be a combination of letters a-z; uppercase and lowercase and numbers 0-9.

The SID prefix cannot have operating system special characters. Therefore, avoid the use of special characters in the first 8 characters of the database unique name. Special characters are omitted if used in the first 8 characters. There is a single SID prefix for every database. The SID prefix for a database must be unique within the cluster.

For an Oracle RAC database, each instance has a unique identifier, ORACLE_SID, which consists of the SID prefix and an instance number. The ORACLE_SID for Oracle RAC database instances is generated differently, depending on how you choose to manage the database. If you select a policy-managed database, then

Oracle generates the SID in the format name_#, where name is the first eight alphanumeric characters of DB_UNIQUE_NAME, and # is the instance number. If you select an admin-managed database, then DBCA generates the SID for the instance names in advance, and the SID is in the format name#.

To find the Sid prefix name, run the following command:

```
# ${GRID_HOME}/bin/crsctl status resource ora.${DBName}.db -f | grep
GEN_USR_ORA_INST_NAME@ | tail -1 | sed 's/.*=//' | sed 's/_[0-9]$//',
```

where *GRID_HOME* is grid home path and *DBName* is the database unique name.

Note: When a policy managed database is created, the Sid prefix is displayed on the confirmation page of the installation procedure.

See "Attribute definitions for the Oracle agent" on page 116.

Resource type definition for the Netlsnr agent

The Netlsnr agent of the Cluster Server agent for Oracle is represented by the Netlsnr resource type in VCS.

```
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 }
 static int IMF{} = { Mode=3, MonitorFreq=5, RegisterRetryLimit=3 }
 static str IMFRegList[] = { Home, Owner, Listener }
 str Owner
 str Home
 str TnsAdmin
 str Listener = "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 Netlsnr agent

Required attributes	Type and dimension	Definition
Owner	string-scalar	The Oracle user who has privileges to start or stop the listener process. 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 Netlsnr agent. You can configure the optional attributes if necessary.

Table A-5 Optional attributes for Netlsnr agent

Optional attributes	Type and dimension	Definition
ContainerOpts	static-assoc-int	This resource-type level attribute specifies the container options for Netlsnr instances running in the context of Solaris containers (zones or projects). These values are only effective when you set the ContainerInfo service group attribute. Veritas recommends that you do not change the following values. ■ RunInContainer (RIC) 1—The Netlsnr agent monitors the listener processes that run inside of the local container. 0—The Netlsnr agent monitors the listener processes that run outside the local container (in the global environment). Default is 1. ■ PassCInfo (PCI) The Netlsnr agent receives the container information that is defined in the service group's ContainerInfo attribute. Default is 0. See the *Cluster Server Administrator's Guide* and the *Veritas InfoScale 7.4.2 Solutions Virtualization Guide*.

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

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 Netlsnr agent and the Oracle database server. Default is LISTENER. **Note:** If a grid infrastructure is set up, make sure that you do not use the default name for the database listener. Otherwise, you may encounter various issues, because the grid infrastructure starts a listener with its default name, which is LISTENER.
LsnrPwd	string-scalar	The VCS 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 VCS Encrypt utility. See "Encrypting Oracle database user and listener passwords" on page 82.
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.

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

Optional attributes	Type and dimension	Definition
IMF	integer-association	This resource-type level attribute determines whether the NetIsnr agent must perform intelligent resource monitoring. You can also override the value of this attribute at resource-level. This attribute includes the following keys: ■ Mode: Define this attribute to enable or disable intelligent resource monitoring. Valid values are as follows: ■ 0—Does not perform intelligent resource monitoring ■ 1—Performs intelligent resource monitoring for offline resources and performs poll-based monitoring for online resources ■ 2—Performs intelligent resource monitoring for online resources and performs poll-based monitoring for offline resources ■ 3—Performs intelligent resource monitoring for both online and for offline resources Default: 3 ■ MonitorFreq: This key value specifies the frequency at which the agent invokes the monitor agent function. The value of this key is an integer. Default: 5 You can set this attribute to a non-zero value in some cases where the agent requires to perform poll-based resource monitoring in addition to the intelligent resource monitoring. After the resource registers with the AMF kernel driver, the agent calls the monitor agent function as follows: ■ After every (MonitorFreq x MonitorInterval) number of seconds for online resources ■ After every (MonitorFreq x OfflineMonitorInterval) number of seconds for offline resources ■ RegisterRetryLimit: If you enable intelligent resource monitoring, the agent invokes the netlsnr_imf_register agent function to register the resource with the AMF kernel driver. The value of the RegisterRetryLimit key determines the number of times the agent must retry registration for a resource. If the agent cannot register the resource within the limit that is specified, then intelligent monitoring is disabled until the resource state changes or the value of the Mode key changes. Default: 3 See "Enabling and disabling intelligent resource monitoring for agents manually" on page 89.

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

Optional attributes	Type and dimension	Definition
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.
LevelTwo MonitorFreq	integer-scalar	Specifies the frequency at which the agent for this resource type must perform second-level or detailed monitoring. If you enabled detail monitoring, then set the value of the LevelTwoMonitorFreq attribute. Default = 0
Encoding	string-scalar	Specifies operating system encoding that corresponds to Oracle encoding for the displayed Oracle output. For example, if Oracle output is in "JAPANESE_JAPAN.JA16EUC," then "eucJP" is the Solaris value for Encoding. Refer to the Oracle and Solaris documentation for respective encoding values. Default is "".
IntentionalOffline		For future use. 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. Veritas recommends not to modify the value of this attribute.

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.

Resource type definition for the ASMInst agent

The ASMInst agent of the Cluster Server agent for Oracle is represented by the ASMInst resource type in VCS.

```
type ASMInst (
 static str AgentDirectory = "/opt/VRTSagents/ha/bin/ASMInst"
 static str ArgList[] = { Sid, Owner, Home, DBAUser,
   DBAPword, Pfile, StartUpOpt, ShutDownOpt,
   EnvFile, Encoding, MonitorOption }
 static int ContainerOpts{} = { RunInContainer=1, PassCInfo=0 }
 str Sid
 str Owner
 str Home
 str DBAUser
 str DBAPword
 str Pfile
 str StartUpOpt
 str ShutDownOpt
 str EnvFile
 str Encoding
 int MonitorOption = 0
)
```

Attribute definition for the ASMInst agent

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

Table A-7 lists the required attributes. You must assign values to the required attributes.

Table A-7 Required attributes for ASMInst agent

Required attributes	Type and dimension	Definition
Sid	string-scalar	The variable $ORACLE_SID that represents the ASM instance. The Sid is considered case-sensitive by the ASMInst agent.
Owner	string-scalar	The Oracle user who has privileges to start or stop ASM instance. The agent also supports LDAP users as Oracle user.

Table A-7 Required attributes for ASMInst agent *(continued)*

Required attributes	Type and dimension	Definition
Home	string-scalar	The $ORACLE_HOME path to Oracle ASM 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-8 lists the optional attributes for ASMInst agent. You can configure the optional attributes if necessary.

Table A-8 Optional attributes for ASMInst agent

Optional Attributes	Type and Dimension	Definition
ContainerOpts	static-assoc-int	This resource-type level attribute specifies the container options for the Oracle ASM instances that run in the context of Solaris containers (zones). These values are only effective when you set the ContainerInfo service group attribute. Veritas recommends that you do not change the following values. ■ RunInContainer (RIC) 1—The ASMInst agent monitors the Oracle ASM instances that run inside of the local container. 0—The ASMInst agent monitors the Oracle ASM instances that run outside the local container (in the global environment). Default is 1. ■ PassCInfo (PCI) The ASMInst agent receives the container information that is defined in the service group's ContainerInfo attribute. Default is 0. See the *Cluster Server Administrator's Guide* and the *Veritas InfoScale 7.4.2 Virtualization Guide*.

Table A-8 Optional attributes for ASMInst agent *(continued)*

Optional Attributes	Type and Dimension	Definition
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 of ASM instance 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 a VCS cluster" on page 198.
DBAUser	string-scalar	The ASM user who has sysasm privileges to start or stop the ASM instance.
DBAPword	string-scalar	Encrypted password for DBAUser. Encrypt passwords only when entering them using the command-line. Passwords must be encrypted using the VCS Encrypt utility. See "Encrypting Oracle database user and listener passwords" on page 82.

Table A-8 Optional attributes for ASMInst agent *(continued)*

Optional Attributes	Type and Dimension	Definition
Encoding	string-scalar	Specifies operating system encoding that corresponds to Oracle encoding for the displayed Oracle output. Default is "".
StartUpOpt	string-scalar	Startup option for the Oracle ASM instance. The agent uses the sqlplus command to start the Oracle ASM instance. This attribute can take the following values: ■ STARTUP ■ STARTUP_MOUNT ■ STARTUP_OPEN ■ SRVCTLSTART ■ SRVCTLSTART_MOUNT ■ SRVCTLSTART_OPEN Default is "STARTUP". **Note:** Veritas recommends to use STARTUP or SRVCTLSTART option for ASMInst resource.
ShutDownOpt	string-scalar	Shut down option for the Oracle ASM instance. By default the agent uses the sqlplus command to stop the Oracle ASM instance. To stop the ASM instance using srvctl utility set the option to SRVCTLSTOP. Default is "".

Table A-9 lists the internal attribute for ASMInst agent. This attribute is for internal use only. Veritas recommends not to modify the value of this attribute.

Table A-9 Internal attributes for ASMInst agent

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

Resource type definition for the ASMDG agent

The ASMDG agent is represented by the ASMDG resource type in VCS. The following extract shows the type definition of the ASMDG resource in the OracleASMTypes.cf file.

```
type ASMDG (
 static str AgentDirectory = "/opt/VRTSagents/ha/bin/ASMDG"
 static str ArgList[] = { Sid, Owner, Home, DBAUser,
   DBAPword, DiskGroups, EnvFile, Encoding }
 static int ContainerOpts{} = { RunInContainer=1, PassCInfo=0 }
 static boolean IntentionalOffline = 1
 str Sid
 str Owner
 str Home
 str DBAUser
 str DBAPword
 keylist DiskGroups
 str EnvFile
 str Encoding
)
```

Attribute definition for the ASMDG agent

The ASMDG agent attributes are classified as required, optional, and internal.

You must assign values to the following required attributes.

Table A-10 Required attributes for ASMDG agent

Required attributes	Type and dimension	Definition
DiskGroups	keylist	The ASM disk groups, where you store the Oracle database files.

Table A-10 Required attributes for ASMDG agent *(continued)*

Required attributes	Type and dimension	Definition
Sid	string-scalar	The variable $ORACLE_SID that represents the ASM instance. The Sid is considered case-sensitive by the ASMInst agent.
Owner	string-scalar	The Oracle user who has privileges to mount or unmount the ASM disk group. The agent also supports LDAP users as Oracle user.
Home	string-scalar	The $ORACLE_HOME path to Oracle ASM 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.

You can configure the following optional attributes if they are necessary for your configuration.

Table A-11 Optional attributes for ASMDG agent

Optional Attributes	Type and Dimension	Definition
DBAUser	string-scalar	The ASM user who has sysasm privileges to start or stop the ASM instance.
DBAPword	string-scalar	Encrypted password for DBAUser. Encrypt passwords only when entering them using the command-line. Passwords must be encrypted using the `vcsencrypt` utility. See "Encrypting Oracle database user and listener passwords" on page 82.

Table A-11 Optional attributes for ASMDG agent *(continued)*

Optional Attributes	Type and Dimension	Definition
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.
Encoding	string-scalar	Specifies operating system encoding that corresponds to Oracle encoding for the displayed Oracle output. Default is "".

The following attribute is for internal use only. Veritas recommends that you do not modify the value of this attribute.

Table A-12 Internal attributes for ASMDG agent

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

Appendix B

Sample configurations

This appendix includes the following topics:

- About the sample configurations for Oracle enterprise agent
- Sample single Oracle instance configuration
- Sample multiple Oracle instances (single listener) configuration
- Sample multiple instance (multiple listeners) configuration
- Sample Oracle configuration with shared server support
- Sample configuration for Oracle instances in Solaris zones
- Sample Oracle ASM configurations
- Sample configuration of Oracle pluggable database (PDB) resource in main.cf
- Sample configuration of migratable Oracle pluggable database (PDB) resource in main.cf
- Sample Configuration of Oracle supported by systemD
- Sample configuration of ASMInst supported by systemD

About the sample configurations for Oracle enterprise agent

The sample configuration includes descriptions for typical service groups that are configured to monitor the state of Oracle in a VCS cluster.

See “Configuring Oracle instances in VCS” on page 68.

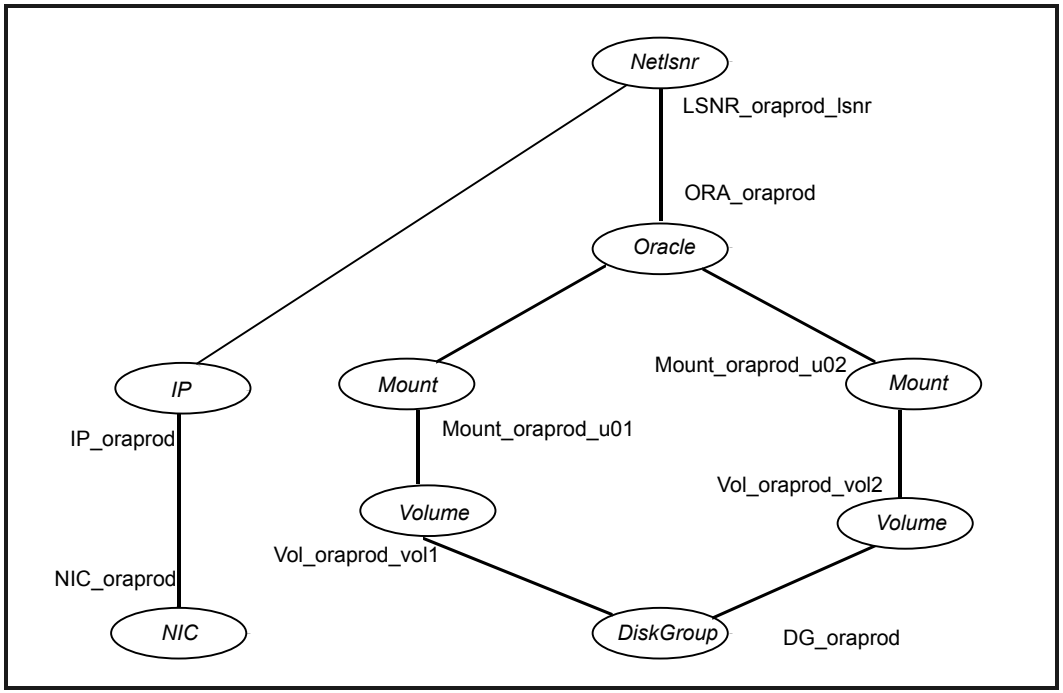
The sample dependency graphs depict the resource types, resources, and resource dependencies within the service group. The sample configuration file (main.cf) is also included for your reference.

Review these dependencies carefully before configuring the agent. For more information about VCS resource types, see the *Cluster Server Bundled Agents Reference Guide*.

Sample single Oracle instance configuration

Figure B-1 describes a typical service group configured to monitor the state of an Oracle instance in a VCS cluster.

Figure B-1 Dependency graph for single Oracle instance



The shared disk groups and volumes in the cluster are configured as resources of type DiskGroup and Volume respectively. The volumes are mounted using the Mount agent. The virtual IP address for the service group is configured using the IP and NIC resource types. The Oracle server can be started after each of these resources is brought online.

Sample VCS configuration file for single Oracle instance

Review the sample configuration with a resource of type Oracle that is configured as follows in main.cf file.

Note: The following sample main.cf uses Solaris SPARC examples for NIC devices. For example, replace the example NIC device from hme0 to bge0 on Solaris x64.

```
include "types.cf"
include "OracleTypes.cf"

cluster vcs (
)

system sys1 (
)

system sys2 (
)

group ORA_PROD_Group (
    SystemList = { sys1 = 0, sys2 = 1 }
    AutoStartList = { sys1 }
)

    DiskGroup DG_oraprod (
        DiskGroup = ora_prod_dg
        StartVolumes = 0
        StopVolumes = 0
    )

    IP IP_oraprod (
        Device = hme0
        Address = "192.168.1.22"
        Netmask = "255.255.255.0"
    )

    Mount Mount_oraprod_u01 (
        MountPoint = "/prod/u01"
        BlockDevice = "/dev/vx/dsk/ora_prod_dg/u01-vol"
        FSType = vxfs
        FsckOpt = "-n"
    )
```

```
Mount Mount_oraprod_u02 (
    MountPoint = "/prod/u02"
    BlockDevice = "/dev/vx/dsk/ora_prod_dg/u02-vol"
    FSType = vxfs
    FsckOpt = "-n"
)

NIC NIC_oraprod (
    Device = hme0
    NetworkHosts = {"192.168.1.1"}
)

Netlsnr LSNR_oraprod_lsnr (
    Owner = oraprod
    Home = "/orahome/Oracle"
    TnsAdmin = "/orahome/Oracle/network/admin"
    Listener = LISTENER_PROD
    MonScript = "./bin/Netlsnr/LsnrTest.pl"
    LsnrPwd = cqfOdoOolOo
)

Oracle ORA_oraprod (
    Sid = PROD
    Owner = oraprod
    Home = "/orahome/Oracle"
    EnvFile = "/tmp/env.sh"
    MonScript = "./bin/Oracle/SqlTest.pl"
    User = thor
    Pword = hvlTptWvj

    DBAUser = “orauser”
    DBAPword = “DPSrFPeRIrGPiRF”

    Table = thor
    MonitorOption = 0
)

Volume Vol_oraprod_vol1 (
    Volume = u01-vol
    DiskGroup = ora_prod_dg
)

Volume Vol_oraprod_vol2 (
```

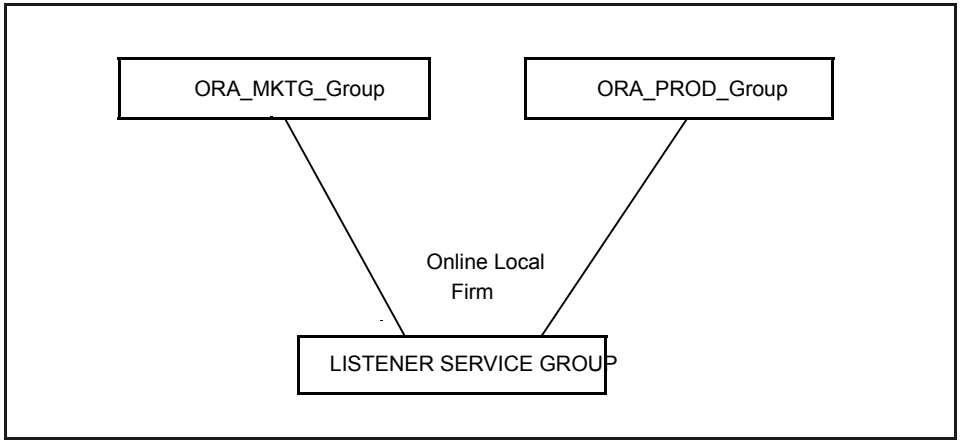
```
    Volume = u02-vol
    DiskGroup = ora_prod_dg
)

IP_oraprod requires NIC_oraprod
LSNR_oraprod_lsnr requires IP_oraprod
LSNR_oraprod_lsnr requires ORA_oraprod
Mount_oraprod_u01 requires Vol_oraprod_vol1
Mount_oraprod_u02 requires Vol_oraprod_vol2
ORA_oraprod requires Mount_oraprod_u01
ORA_oraprod requires Mount_oraprod_u02
Vol_oraprod_vol1 requires DG_oraprod
Vol_oraprod_vol2 requires DG_oraprod
```

Sample multiple Oracle instances (single listener) configuration

Figure B-2 describes a typical VCS configuration to monitor two Oracle instances sharing a listener. This configuration has a service group for each Oracle instance. The Listener too is configured in a separate service group.

Figure B-2 Two Oracle instances sharing a listener



The Oracle service groups are made dependent on the Listener service group using an Online Local Firm dependency.

Figure B-3 shows the dependency graph for one of the Oracle instances in the VCS configuration. In the Oracle service group, the shared disk groups and volumes in

the cluster are configured as resources of type DiskGroup and Volume respectively. The volumes are mounted using the Mount agent.

Figure B-3 Dependency graph for one of the Oracle instances

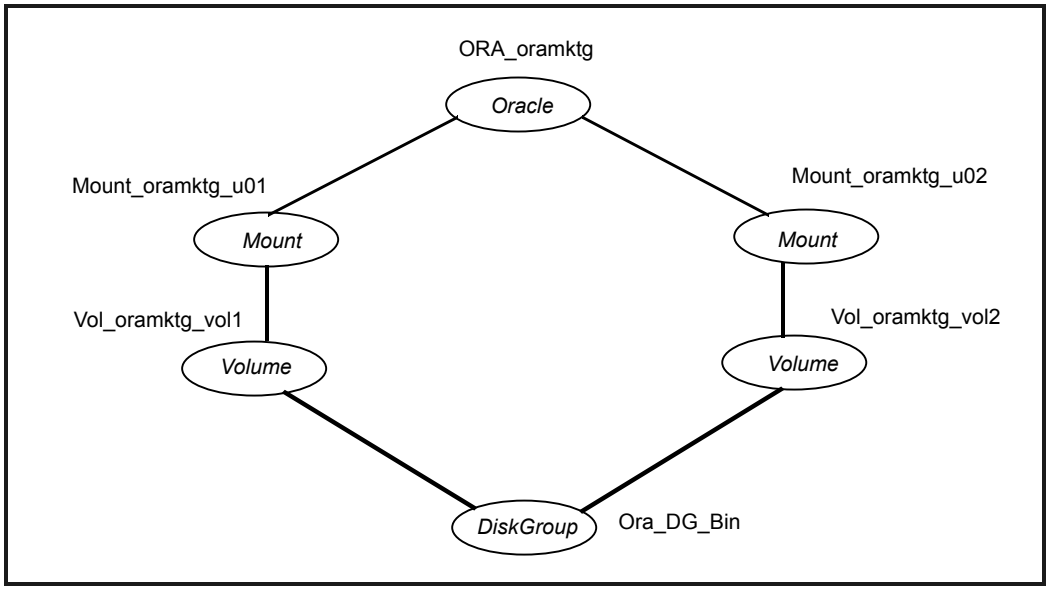
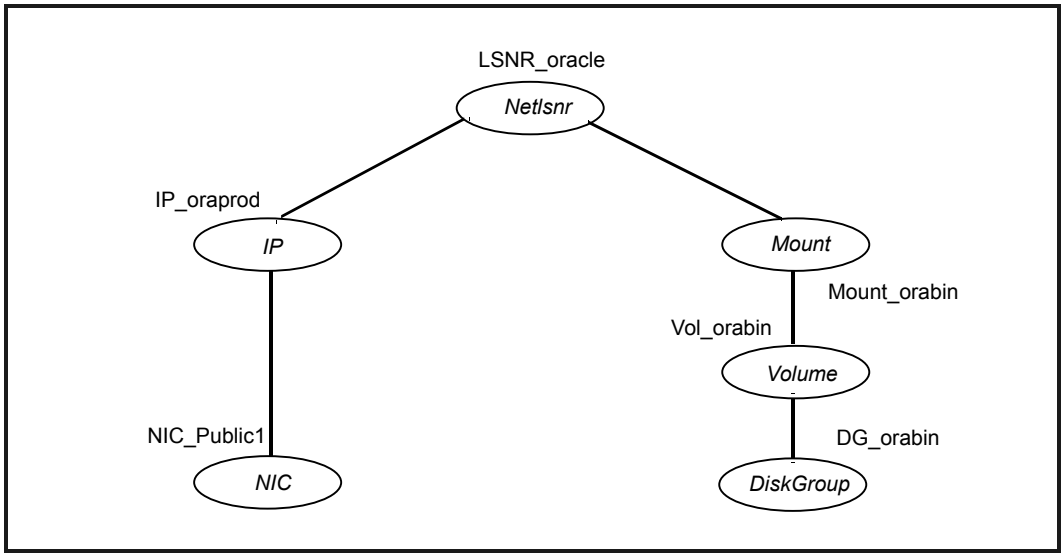


Figure B-4 shows the dependency graph for the listener that the two Oracle instances share in the VCS configuration. In the Listener service group, the virtual IP address is configured using the IP and NIC resource types. The Listener can be started after the IP and NIC resources are brought online.

Figure B-4 Dependency graph for the single listener



The Oracle server can be started after the Listener service group and the resources in the Oracle service group are brought online.

Note: In this case, make sure you have modified all proper system files, such as /etc/system, /etc/passwd, /etc/group, and /etc/shadow to support multiple databases. Pay particular attention to system requirements like physical memory and shared memory segment availability. Also ensure a single system is capable of sustaining a multiple instance load in the event of a server failure and extended operation on the backup server.

Sample VCS configuration file for multiple Oracle instances (single listener)

Review the sample configuration file for multiple Oracle instances.

Note: The following sample main.cf uses Solaris SPARC examples for NIC devices. For example, replace the example NIC device from hme0 to bge0 on Solaris x64.

```
include "types.cf"
include "OracleTypes.cf"

cluster vcs (
```

```
)
system sys1 (
)

system sys2 (
)

group ORA_MKTG_Group (
    SystemList = { sys1 = 0, sys2 = 1 }
    AutoStartList = { sys1 }
)

    DiskGroup DG_oramktg (
        DiskGroup = ora_mktg_dg
        StartVolumes = 0
        StopVolumes = 0
    )

    Mount Mount_oramktg_u01 (
        MountPoint = "/mktg/u01"
        BlockDevice = "/dev/vx/dsk/ora_mktg_dg/u01-vol"
        FSType = vxfs
        FsckOpt = "-n"
    )

    Mount Mount_oramktg_u02 (
        MountPoint = "/mktg/u02"
        BlockDevice = "/dev/vx/dsk/ora_mktg_dg/u02-vol"
        FSType = vxfs
        FsckOpt = "-n"
    )

    Oracle ORA_oramktg (
        Sid = MKTG
        Owner = oramktg
        Home = "/orahome/Oracle"
        EnvFile = "/tmp/env.sh"
        MonScript = "./bin/Oracle/SqlTest.pl"
        User = thor
        Pword = hvlTptWvj
        Table = thor
        MonitorOption = 0
    )
```

```
    Volume Vol_oramktg_vol1 (
        Volume = u01-vol
        DiskGroup = ora_mktg_dg
    )

    Volume Vol_oramktg_vol2 (
        Volume = u02-vol
        DiskGroup = ora_mktg_dg
    )

    requires group Common_Service online local firm
    Mount_oramktg_u01 requires Vol_oramktg_vol1
    Mount_oramktg_u02 requires Vol_oramktg_vol2
    ORA_oramktg requires Mount_oramktg_u01
    ORA_oramktg requires Mount_oramktg_u02
    Vol_oramktg_vol1 requires DG_oramktg
    Vol_oramktg_vol2 requires DG_oramktg


group ORA_PROD_Group (
    SystemList = { sys1 = 0, sys2 = 1 }
    AutoStartList = { sys1 }
)

    DiskGroup DG_oraprod (
        DiskGroup = ora_prod_dg
        StartVolumes = 0
        StopVolumes = 0
    )

    Mount Mount_oraprod_u01 (
        MountPoint = "/prod/u01"
        BlockDevice = "/dev/vx/dsk/ora_prod_dg/u01-vol"
        FSType = vxfs
        FsckOpt = "-n"
    )

    Mount Mount_oraprod_u02 (
        MountPoint = "/prod/u02"
        BlockDevice = "/dev/vx/dsk/ora_prod_dg/u02-vol"
        FSType = vxfs
        FsckOpt = "-n"
```

```
    )

    Oracle ORA_oraprod (
        Sid = PROD
        Owner = oraprod
        Home = "/orahome/Oracle"
        EnvFile = "/tmp/env.sh"
        MonScript = "./bin/Oracle/SqlTest.pl"
        User = thor
        Pword = hvlTptWvj
        Table = thor
        MonitorOption = 0
    )

    Volume Vol_oraprod_vol1 (
        Volume = u01-vol
        DiskGroup = ora_prod_dg
    )

    Volume Vol_oraprod_vol2 (
        Volume = u02-vol
        DiskGroup = ora_prod_dg
    )

    requires group Common_Service online local firm
    Mount_oraprod_u01 requires Vol_oraprod_vol1
    Mount_oraprod_u02 requires Vol_oraprod_vol2
    ORA_oraprod requires Mount_oraprod_u01
    ORA_oraprod requires Mount_oraprod_u02
    Vol_oraprod_vol1 requires DG_oraprod
    Vol_oraprod_vol2 requires DG_oraprod


group Common_Service (
    SystemList = { sys1 = 0, sys2 = 1 }
    AutoStartList = { sys1 }
)

    DiskGroup DG_orabin (
        DiskGroup = ora_bin_dg
        StartVolumes = 0
        StopVolumes = 0
    )
```

```
IP IP_oraprod (
    Device = hme0
    Address = "192.168.1.22"
    Netmask = "255.255.255.0"
)

Mount Mount_orabin (
    MountPoint = "/orahome/Oracle"
    BlockDevice = "/dev/vx/dsk/ora_bin_dg/u01-vol"
    FSType = vxfs
    FsckOpt = "-n"
)

NIC NIC_Public1 (
    Device = hme0
    NetworkHosts = {"192.168.1.1"}
)

Netlsnr LSNR_oracle (
    Owner = oracle
    Home = "/orahome/Oracle"
    TnsAdmin = "/orahome/Oracle/network/admin"
    Listener = LISTENER_ORACLE
    MonScript = "./bin/Netlsnr/LsnrTest.pl"
)

Volume Vol_orabin (
    Volume = u01-vol
    DiskGroup = ora_bin_dg
)

IP_oraprod requires NIC_Public1
LSNR_oracle requires IP_oraprod
LSNR_oracle requires Mount_orabin
Mount_orabin requires Vol_orabin
Vol_orabin requires DG_orabin
```

Sample multiple instance (multiple listeners) configuration

This configuration has several single-instance configurations. Each Oracle instance is configured in a separate service group. The resource dependencies are similar to the single Oracle instance configuration.

See "Sample single Oracle instance configuration" on page 140.

Sample VCS configuration file for multiple Oracle instances (multiple listeners)

Review the sample configuration file for the multiple Oracle instances that have multiple listeners.

Note: The following sample main.cf uses Solaris SPARC examples for NIC devices. For example, replace the example NIC device from hme0 to bge0 on Solaris x64.

```
include "types.cf"
include "OracleTypes.cf"

cluster vcs (
)

system sys1 (
)

system sys2 (
)

group ORA_MKTG_Group (
    SystemList = { sys1 = 0, sys2 = 1 }
    AutoStartList = { sys1 }
)

    DiskGroup DG_oramktg (
        DiskGroup = ora_mktg_dg
        StartVolumes = 0
        StopVolumes = 0
    )

    IP IP_oramktg (
```

```
    Device = hme0
    Address = "192.168.1.22"
    Netmask = "255.255.255.0"
)

Mount Mount_oramktg_u01 (
    MountPoint = "/mktg/u01"
    BlockDevice = "/dev/vx/dsk/ora_mktg_dg/u01-vol"
    FSType = vxfs
    FsckOpt = "-n"
)

Mount Mount_oramktg_u02 (
    MountPoint = "/mktg/u02"
    BlockDevice = "/dev/vx/dsk/ora_mktg_dg/u02-vol"
    FSType = vxfs
    FsckOpt = "-n"
)

Netlsnr LSNR_oramktg_lsnr (
    Owner = oramktg
    Home = "/orahome/Oracle"
    TnsAdmin = "/orahome/Oracle/network/admin"
    Listener = LISTENER_MKTG
    MonScript = "./bin/Netlsnr/LsnrTest.pl"
    LsnrPwd = cqfOdoOolOo
)

Oracle ORA_oramktg (
    Sid = MKTG
    Owner = oramktg
    Home = "/orahome/Oracle"
    EnvFile = "/tmp/env.sh"
    MonScript = "./bin/Oracle/SqlTest.pl"
    User = thor
    Pword = hvlTptWvj
    Table = thor
    MonitorOption = 0
)

Proxy NICProxy_oramktg (
    TargetResName = NIC_Public1
)
```

```
    Volume Vol_oramktg_vol1 (
        Volume = u01-vol
        DiskGroup = ora_mktg_dg
    )

    Volume Vol_oramktg_vol2 (
        Volume = u02-vol
        DiskGroup = ora_mktg_dg
    )

    IP_oramktg requires NICProxy_oramktg
    LSNR_oramktg_lsnr requires IP_oramktg
    LSNR_oramktg_lsnr requires ORA_oramktg
    Mount_oramktg_u01 requires Vol_oramktg_vol1
    Mount_oramktg_u02 requires Vol_oramktg_vol2
    ORA_oramktg requires Mount_oramktg_u01
    ORA_oramktg requires Mount_oramktg_u02
    Vol_oramktg_vol1 requires DG_oramktg
    Vol_oramktg_vol2 requires DG_oramktg


group ORA_PROD_Group (
    SystemList = { sys1 = 0, sys2 = 1 }
    AutoStartList = { sys1 }
)
    DiskGroup DG_oraprod (
        DiskGroup = ora_prod_dg
        StartVolumes = 0
        StopVolumes = 0
    )

    IP IP_oraprod (
        Device = hme0
        Address = "192.168.1.21"
        Netmask = "255.255.255.0"
    )

    Mount Mount_oraprod_u01 (
        MountPoint = "/prod/u01"
        BlockDevice = "/dev/vx/dsk/ora_prod_dg/u01-vol"
        FSType = vxfs
        FsckOpt = "-n"
```

```
)

Mount Mount_oraprod_u02 (
    MountPoint = "/prod/u02"
    BlockDevice = "/dev/vx/dsk/ora_prod_dg/u02-vol"
    FSType = vxfs
    FsckOpt = "-n"
)

Netlsnr LSNR_oraprod_lsnr (
    Owner = oraprod
    Home = "/orahome/Oracle"
    TnsAdmin = "/orahome/Oracle/network/admin"
    Listener = LISTENER_PROD
    MonScript = "./bin/Netlsnr/LsnrTest.pl"
    LsnrPwd = cqfOdoOolOo
)

Oracle ORA_oraprod (
    Sid = PROD
    Owner = oraprod
    Home = "/orahome/Oracle"
    EnvFile = "/tmp/env.sh"
    MonScript = "./bin/Oracle/SqlTest.pl"
    User = thor
    Pword = hvlTptWvj
    Table = thor
    MonitorOption = 0
)

Proxy NICProxy_oraprod (
    TargetResName = NIC_Public1
)

Volume Vol_oraprod_vol1 (
    Volume = u01-vol
    DiskGroup = ora_prod_dg
)

Volume Vol_oraprod_vol2 (
    Volume = u02-vol
    DiskGroup = ora_prod_dg
)
```

```
    IP_oraprod requires NICProxy_oraprod
    LSNR_oraprod_lsnr requires IP_oraprod
    LSNR_oraprod_lsnr requires ORA_oraprod
    Mount_oraprod_u01 requires Vol_oraprod_vol1
    Mount_oraprod_u02 requires Vol_oraprod_vol2
    ORA_oraprod requires Mount_oraprod_u01
    ORA_oraprod requires Mount_oraprod_u02
    Vol_oraprod_vol1 requires DG_oraprod
    Vol_oraprod_vol2 requires DG_oraprod


group Parallel_Service (
    SystemList = { sys1 = 0, sys2 = 1 }
    Parallel = 1
    AutoStartList = { sys1 }
)

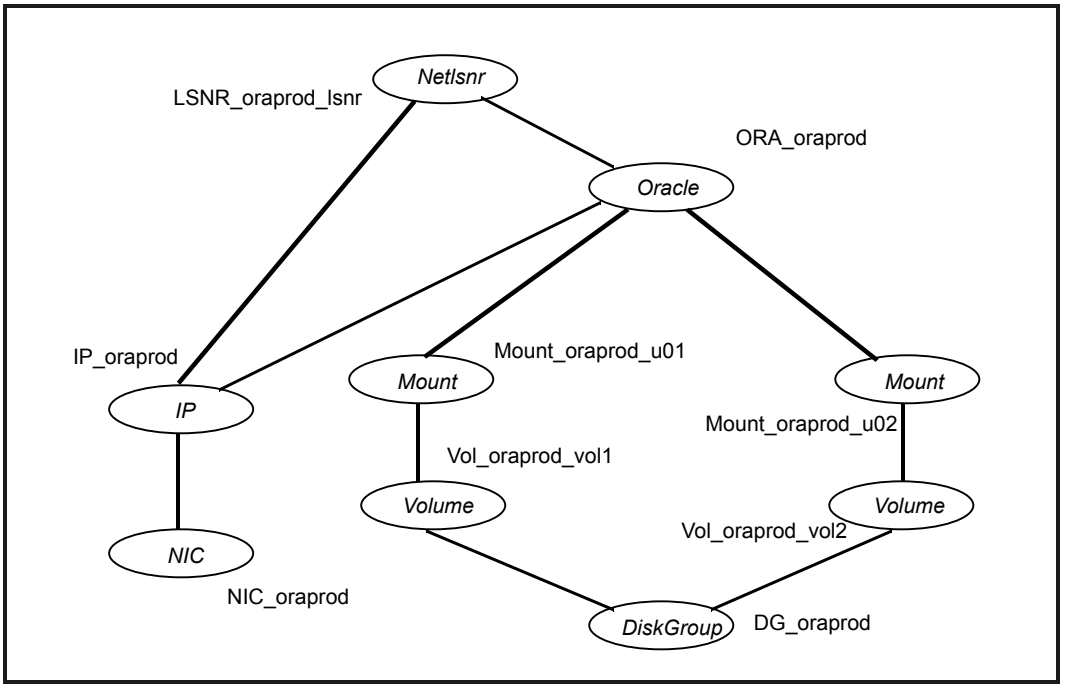
    NIC NIC_Public1 (
        Device = hme0
        NetworkHosts = {"192.168.1.1"}
    )
```

Sample Oracle configuration with shared server support

Figure B-5 describes a typical service group configured to monitor Oracle with shared server support.

You can also configure Oracle instances with shared server support in Solaris zones.

Figure B-5 Dependency for Oracle configured with shared server support



The shared disk groups and volumes in the cluster are configured as resources of type DiskGroup and Volume respectively. The volumes are mounted using the Mount agent. The virtual IP address for the service group is configured using the IP and NIC resource types. The Oracle server can be started after each of these resources is brought online.

Sample VCS configuration file for Oracle instance configured with shared server support

Review the configuration file for an Oracle instance that is configured with shared server support.

Note: The following sample main.cf uses Solaris SPARC examples for NIC devices. For example, replace the example NIC device from hme0 to bge0 on Solaris x64.

```
include "types.cf"
include "OracleTypes.cf"
```

```
cluster vcs (
)

system sys1 (
)

system sys2 (
)

group ORA_PROD_Group (
    SystemList = { sys1 = 0, sys2 = 1 }
    AutoStartList = { sys1 }
)

    DiskGroup DG_oraprod (
        DiskGroup = ora_prod_dg
        StartVolumes = 0
        StopVolumes = 0
    )

    IP IP_oraprod (
        Device = hme0
        Address = "192.168.1.22"
        Netmask = "255.255.255.0"
    )

    Mount Mount_oraprod_u01 (
        MountPoint = "/prod/u01"
        BlockDevice = "/dev/vx/dsk/ora_prod_dg/u01-vol"
        FSType = vxfs
        FsckOpt = "-n"
    )

    Mount Mount_oraprod_u02 (
        MountPoint = "/prod/u02"
        BlockDevice = "/dev/vx/dsk/ora_prod_dg/u02-vol"
        FSType = vxfs
        FsckOpt = "-n"
    )

    NIC NIC_ORAPROD (
        Device = hme0
        NetworkHosts = {"192.168.1.1"}
```

```
)

Netlsnr LSNR_oraprod_lsnr (
    Owner = oraprod
    Home = "/orahome/Oracle"
    TnsAdmin = "/orahome/Oracle/network/admin"
    Listener = LISTENER_PROD
    MonScript = "./bin/Netlsnr/LsnrTest.pl"
    LsnrPwd = cqfOdoOolOo
)

Oracle ORA_oraprod (
    Sid = PROD
    Owner = oraprod
    Home = "/orahome/Oracle"
    EnvFile = "/tmp/env.sh"
    MonScript = "./bin/Oracle/SqlTest.pl"
    User = thor
    Pword = hvlTptWvj
    Table = thor
    MonitorOption = 0
)

Volume Vol_oraprod_vol1 (
    Volume = u01-vol
    DiskGroup = ora_prod_dg
)

Volume Vol_oraprod_vol2 (
    Volume = u02-vol
    DiskGroup = ora_prod_dg
)

IP_oraprod requires NIC_ORAPROD
LSNR_oraprod_lsnr requires IP_oraprod
Mount_oraprod_u01 requires Vol_oraprod_vol1
Mount_oraprod_u02 requires Vol_oraprod_vol2
ORA_oraprod requires IP_oraprod
ORA_oraprod requires Mount_oraprod_u01
ORA_oraprod requires Mount_oraprod_u02
Vol_oraprod_vol1 requires DG_oraprod
Vol_oraprod_vol2 requires DG_oraprod
```

Sample configuration for Oracle instances in Solaris zones

You can configure Oracle instances in a Solaris zone with the zone root on either the local disk or the shared disk.

Review the following dependency charts for the different Oracle configurations in Solaris zones that VCS supports:

- Single Oracle instance configuration
  See "Zone root on local disk for single Oracle instance" on page 158.
  See "Zone root on shared disk for single Oracle instance" on page 159.
- Multiple Oracle instances (single listener) configuration
  See "Zone root on local disk for multiple Oracle instances" on page 160.
  See "Zone root on shared disk for multiple Oracle instances" on page 163.
- Multiple Oracle instances (multiple listeners) configuration
- Oracle configuration with shared server support
  See "Zone root on local disk for Oracle instance with shared server support" on page 165.
  See "Zone root on shared disk for Oracle instance with shared server support" on page 166.

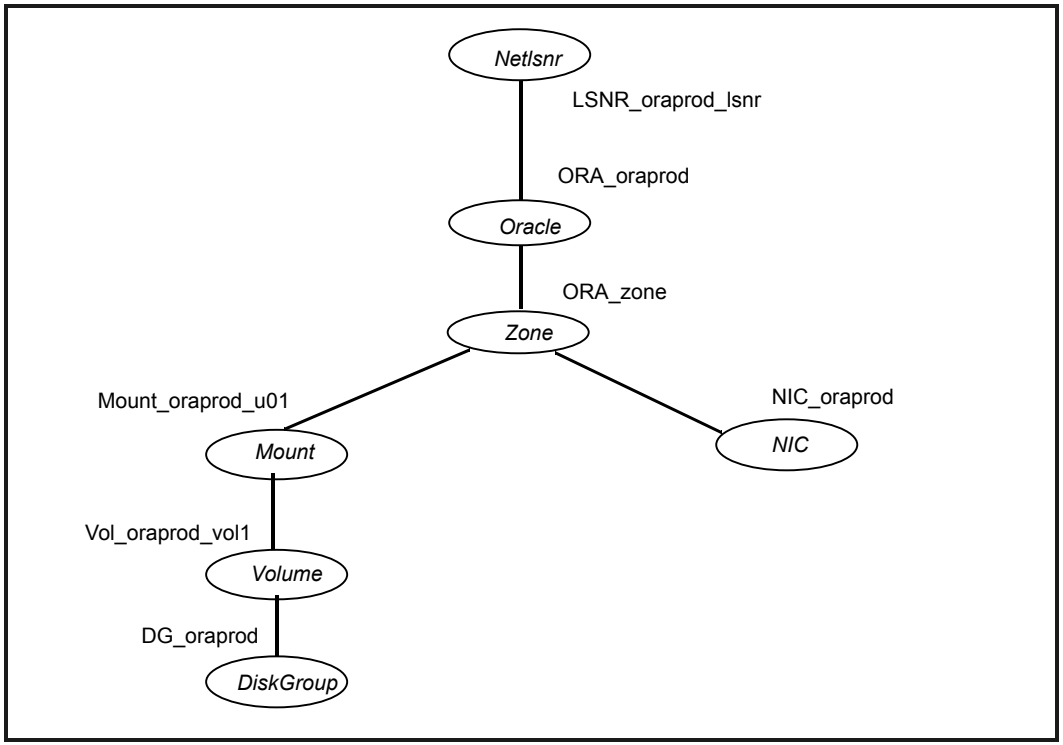
See the *Veritas InfoScale7.4.2 Virtualization Guide* for dependency chart examples with direct mount points.

Review the VCS sample configuration file for Oracle instances in Solaris zones.

Zone root on local disk for single Oracle instance

Figure B-6 describes a typical service group in a Solaris zone that is on local disk. If the root file system of a zone is on the local disk of each node, the file system is mounted when the system is booted. Hence, the service group does not need to have separate DiskGroup and Volume resources for the zone.

Figure B-6 Single Oracle instance in zone on local disk

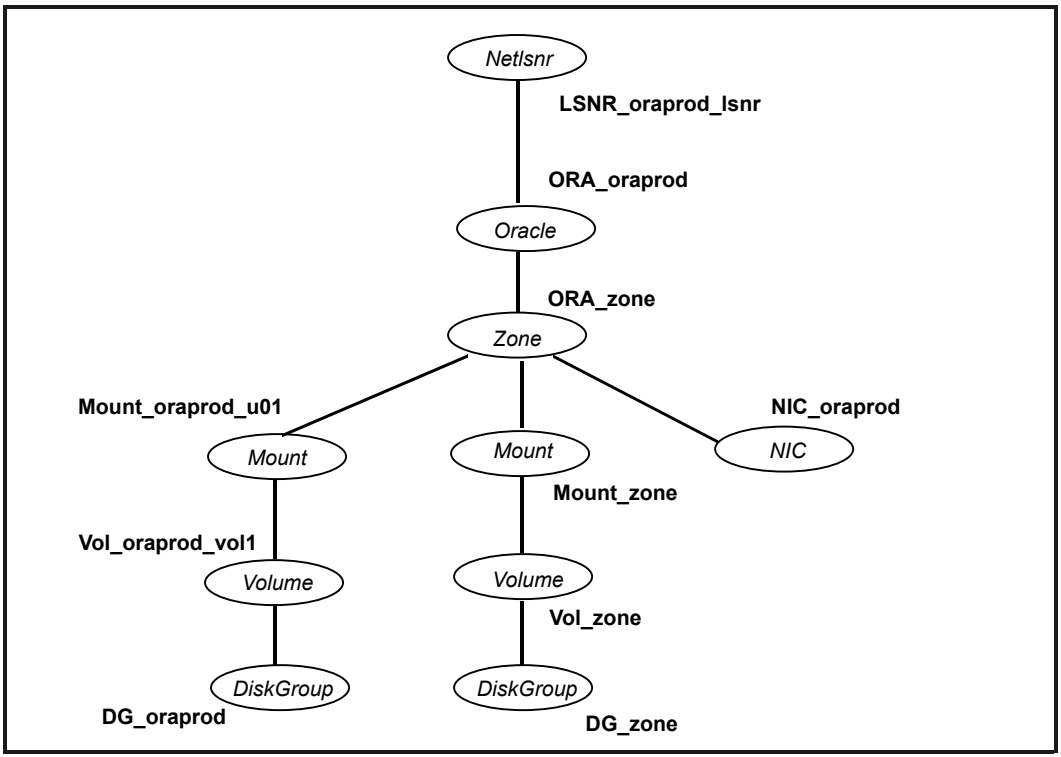


In this sample configuration Oracle is installed locally inside the zone and the data files of Oracle instance are stored on shared disks. The shared disk groups and volumes in the cluster are configured as resources of type DiskGroup and Volume respectively. The volumes are mounted using the Mount agent. The file system mounted by Mount agent is mounted inside the zone as LOFS mount by zone configuration. The Solaris zone is monitored through a zone resource, which is dependent on the Mount and NIC resources. The Oracle server can be started after each of these resources is brought online.

Zone root on shared disk for single Oracle instance

Figure B-7 describes a typical service group in a Solaris zone that is on shared disk. If the root file system of a zone is on a shared disk, the file system should be mounted by VCS. Hence, separate DiskGroup and Volume resources are required for the zone.

Figure B-7 Single Oracle instance in zone on shared disk



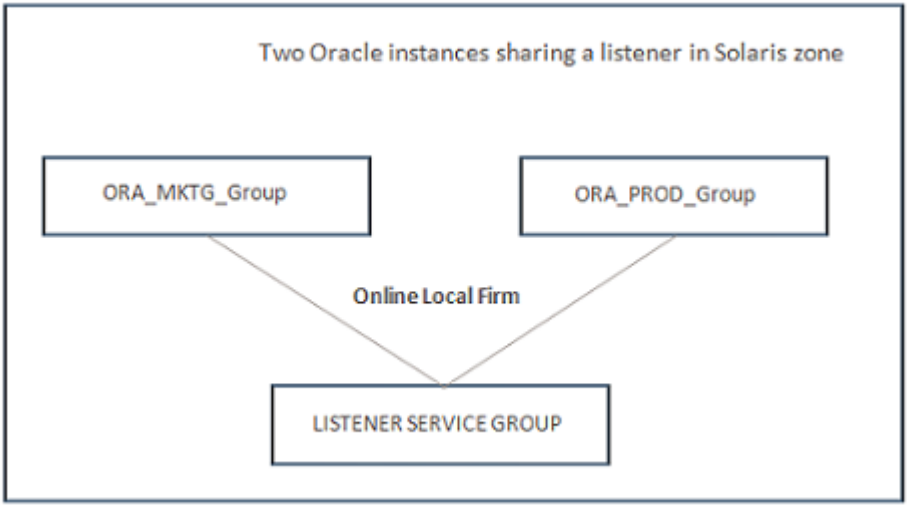
8

In this sample configuration, Oracle Home is installed locally inside the zone and the data files of Oracle instance are stored on shared disks. The shared disk groups and volumes in the cluster are configured as resources of type DiskGroup and Volume respectively. The volumes are mounted using the Mount agent. The file system mounted by Mount agent is mounted inside the zone as LOFS mount by zone configuration. The Solaris zone is monitored through a zone resource, which is dependent on the Mount and NIC resources. The Oracle server can be started after each of these resources is brought online.

Zone root on local disk for multiple Oracle instances

In this configuration the Oracle service groups are made dependent on Netlsnr service group using an Online Local Firm dependency. The zone resource is configured in the Listener service group. The Oracle service groups sets the ContainerInfo attribute for the Zone, but does not contain the zone resource.

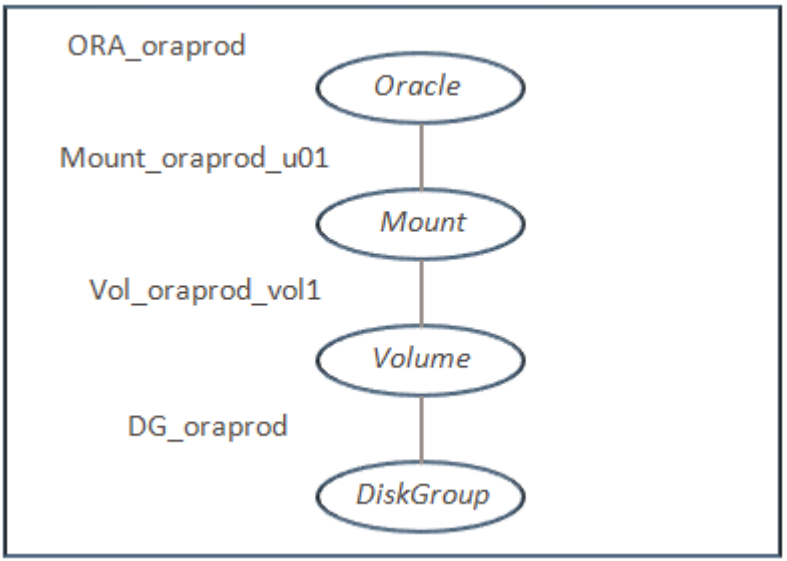
Figure B-8 Two Oracle instances sharing a listener in Solaris zone



If the root file system of a zone is on the local disk of each node, the file system is mounted when the system is booted. Hence, the service group does not need to have separate DiskGroup and Volume resources for the zone.

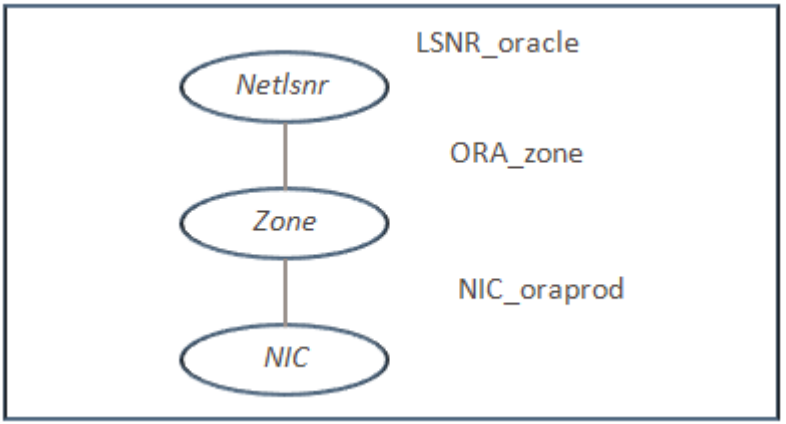
Oracle Home is installed locally inside the zone and the data files of Oracle instance are stored on shared disks. In the Oracle service group, the shared disk groups and volumes in the cluster are configured as resources of type DiskGroup and Volume respectively. The volumes are mounted using the Mount agent. The Solaris zone is monitored through a zone resource, which is dependent on the Mount and NIC resources.

Figure B-9 Dependency for an Oracle service group



In the Listener service group, the Listener resource is dependent on the zone resource. The Listener can be started after the zone resource is brought online.

Figure B-10 Dependency for Netlnsr service group



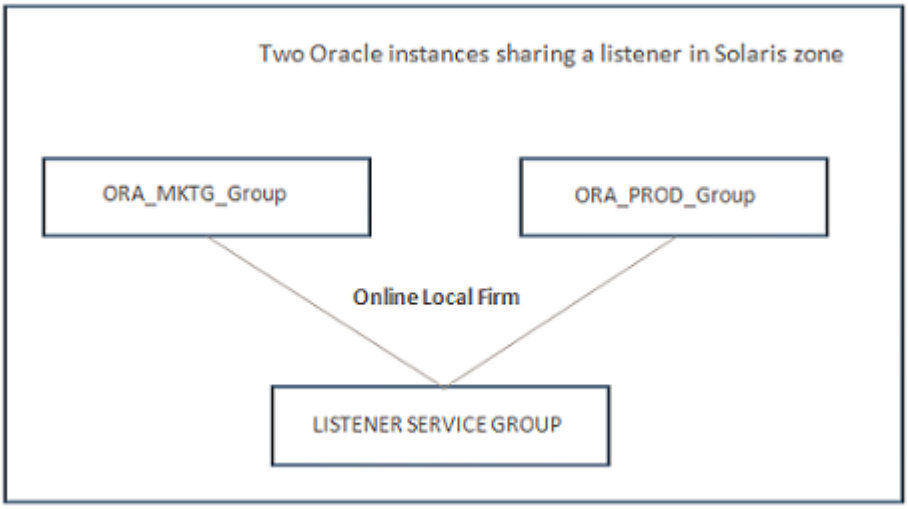
The Oracle server can be started after the Listener service group and the resources in the Oracle service group are brought online.

Note: In this case, make sure you have modified all proper system files, such as /etc/system, /etc/passwd, /etc/group, and /etc/shadow to support multiple databases. Pay particular attention to system requirements like physical memory and shared memory segment availability. Also ensure a single system is capable of sustaining a multiple instance load in the event of a server failure and extended operation on the backup server.

Zone root on shared disk for multiple Oracle instances

In this configuration the Oracle service groups are made dependent on Netlsnr service group using an Online Local Firm dependency. The zone resource is configured in the Listener service group. The Oracle service groups set the ContainerInfo attribute for the Zone, but do not contain the zone resource.

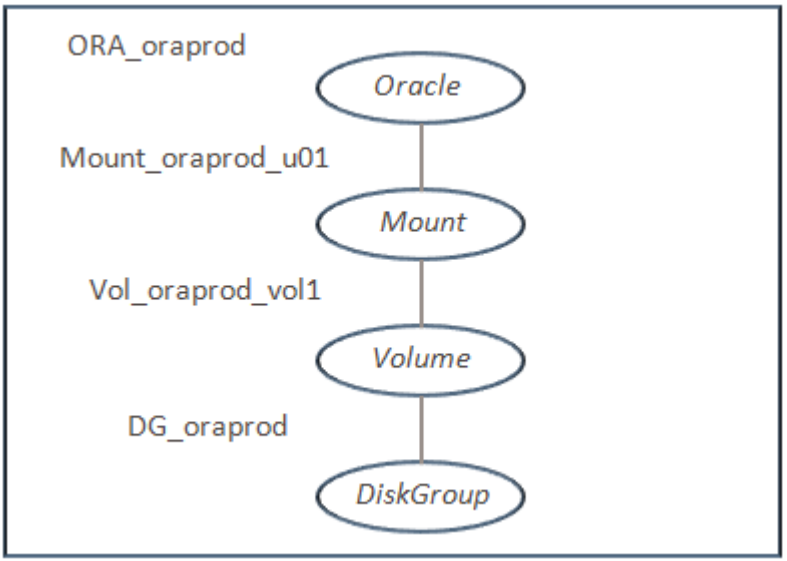
Figure B-11 Two Oracle instances sharing a listener in Solaris zone



If the root file system of a zone is on a shared disk, the file system should be mounted by VCS. Hence, separate DiskGroup and Volume resources are required for the zone.

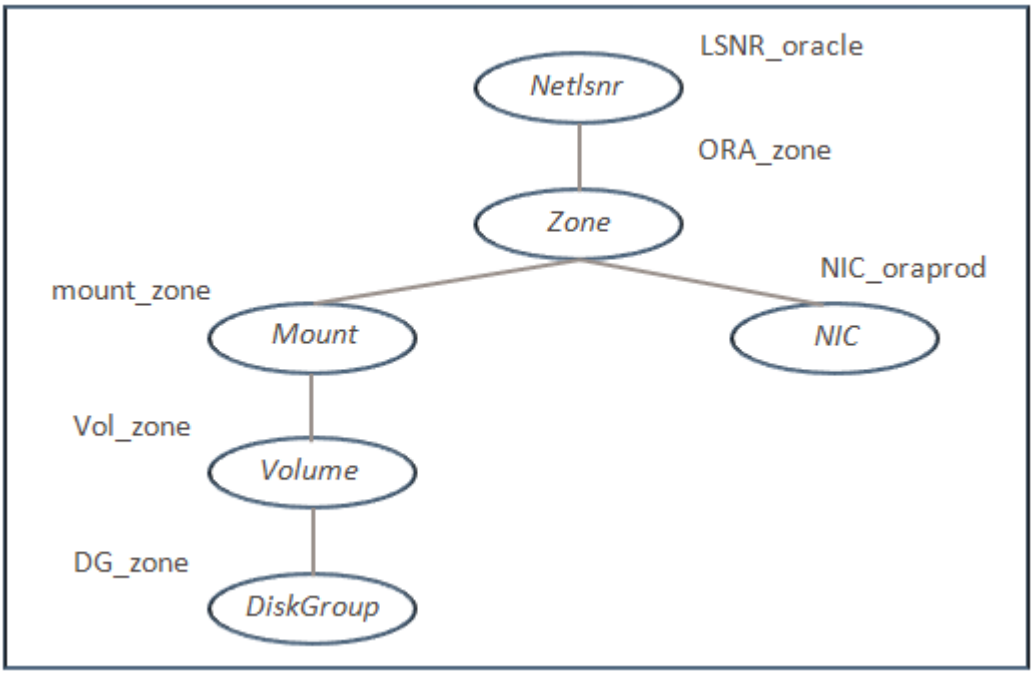
Oracle Home is installed locally inside the zone and the data files of Oracle instance are stored on shared disks. In the Oracle service group, the shared disk groups and volumes in the cluster are configured as resources of type DiskGroup and Volume respectively. The volumes are mounted using the Mount agent. The Solaris zone is monitored through a zone resource, which is dependent on the Mount and NIC resources.

Figure B-12 Dependency for an Oracle service group



In the Listener service group, the Listener resource is dependent on the zone resource. The Listener can be started after the zone resource is brought online.

Figure B-13 Dependency for Netlsnr service group



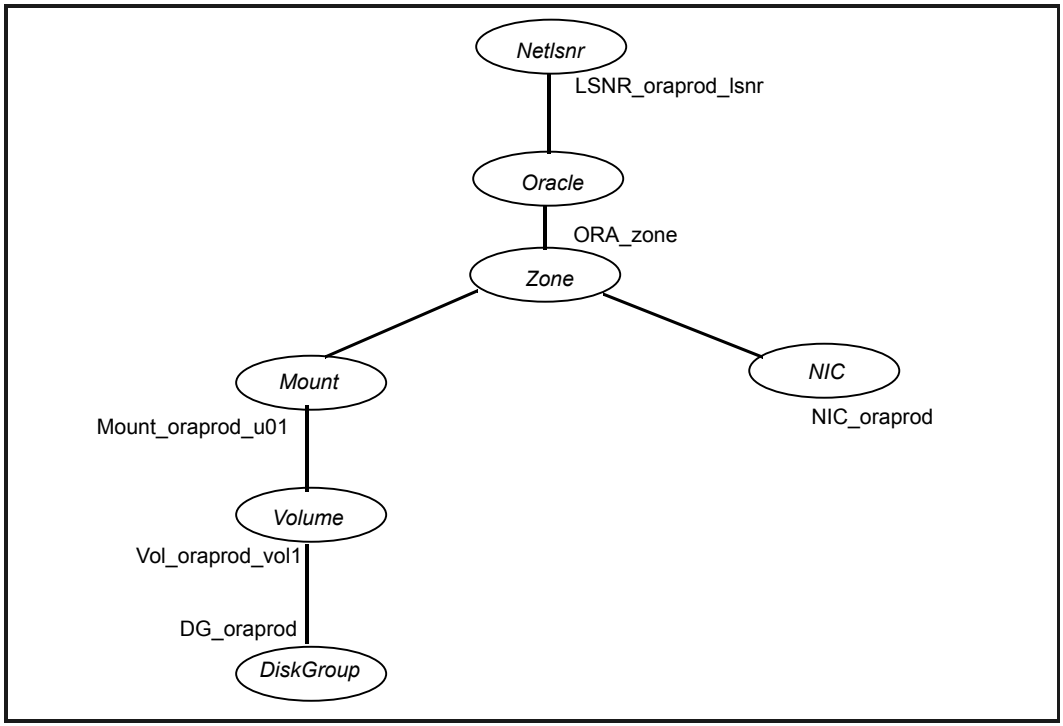
The Oracle server can be started after the Listener service group and the resources in the Oracle service group are brought online.

Note: In this case, make sure you have modified all proper system files, such as /etc/system, /etc/passwd, /etc/group, and /etc/shadow to support multiple databases. Pay particular attention to system requirements like physical memory and shared memory segment availability. Also ensure a single system is capable of sustaining a multiple instance load in the event of a server failure and extended operation on the backup server.

Zone root on local disk for Oracle instance with shared server support

If the root file system of a zone is on the local disk of each node, the file system is mounted when the system is booted. Hence, the service group does not need to have separate DiskGroup and Volume resources for the zone.

Figure B-14 Oracle instance with shared server support in zone on local disk

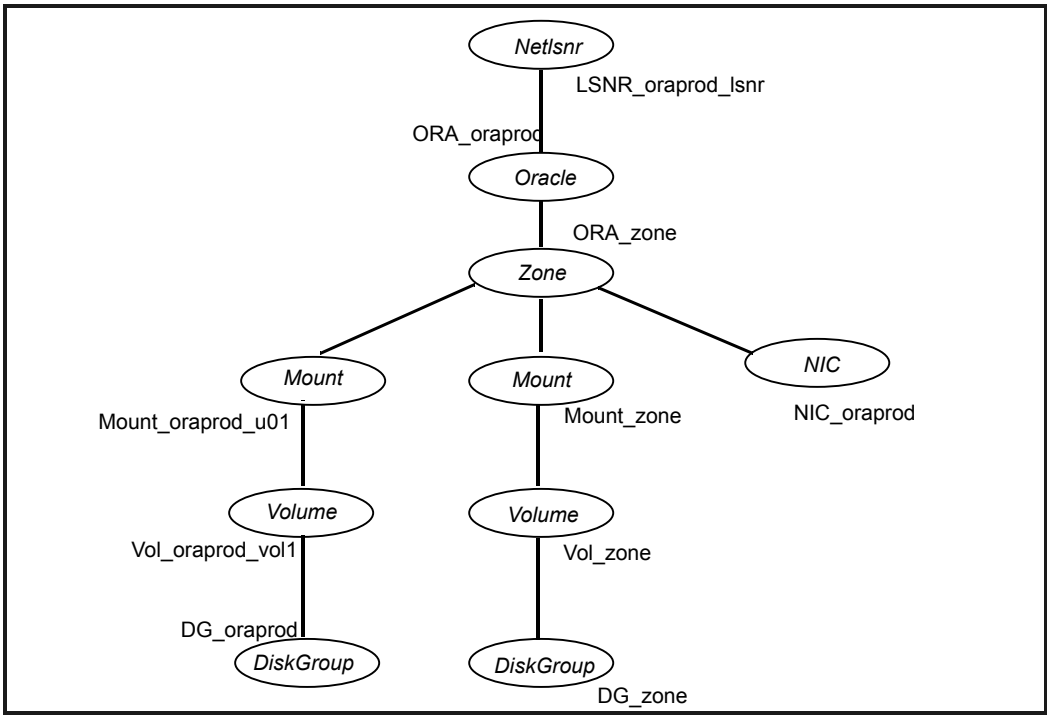


In this sample configuration Oracle Home is installed locally inside the zone and the data files of Oracle instance are stored on shared disks. The shared disk groups and volumes in the cluster are configured as resources of type DiskGroup and Volume respectively. The volumes are mounted using the Mount agent. The file system, mounted by Mount agent, is mounted inside the zone as LOFS mount by zone configuration. The Solaris zone is monitored through a zone resource, which is dependent on the Mount and NIC resources. The Oracle server can be started after each of these resources is brought online.

Zone root on shared disk for Oracle instance with shared server support

If the root file system of a zone is on a shared disk, the file system should be mounted by VCS. Hence, separate DiskGroup and Volume resources are required for the zone.

Figure B-15 Oracle instance with shared server support in zone on shared disk

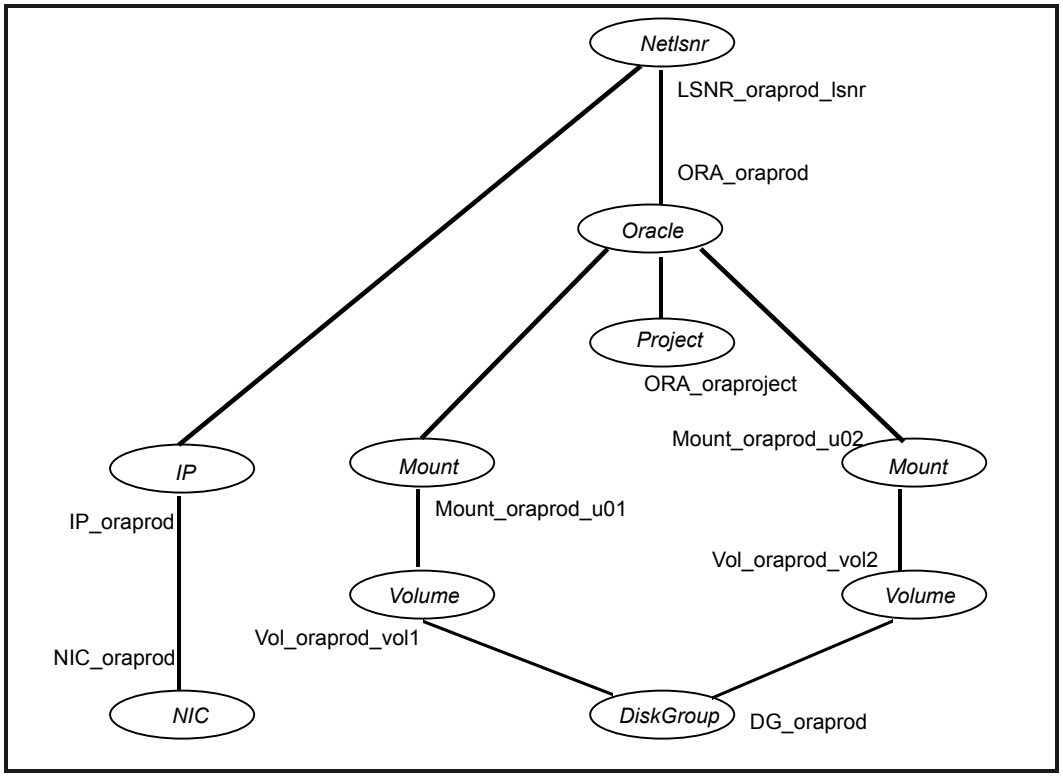


The shared disk groups and volumes in the cluster are configured as resources of type DiskGroup and Volume respectively. The volumes are mounted using the Mount agent. The Solaris zone is monitored through a zone resource, which is dependent on the Mount and NIC resources. The Oracle server can be started after each of these resources is brought online.

Oracle instance in the context of project

Figure B-16 describes a typical service group configured to monitor the state of an Oracle instance running in the context of a Solaris project.

Figure B-16 Single Oracle instance running in the context of Solaris project



The Solaris project is monitored through a project resource.

Sample VCS configuration file for zone root on local disk or on shared disk

A resource of type Zone may be configured as follows in the main.cf:

```
Zone ORA_zone (
)
```

Along with this resource addition, the service group configuration would also change to have the NIC and Mount resources to depend on the Zone resource. Refer to the resource dependency diagrams to modify the resource dependencies to reflect your zone configuration.

See “Sample configuration for Oracle instances in Solaris zones” on page 158.

Verify that the value of the group-level attribute ContainerInfo attribute is defined correctly for the following keys:

- Name: Define the name of the non-global zone
- Type: Set this value to Zone.
- Enabled: Set this value to 0 or 1 depending on how you want to manage the resources.

See the *Cluster Server Administrator's Guide* and the *Veritas InfoScale 7.4.2 Virtualization Guide*.

Sample Oracle ASM configurations

Review the dependency graphs for the service group configured to monitor the state of an Oracle instance that is ASM-managed in a VCS cluster. You can have the following Oracle ASM configurations in a VCS environment:

- ASM disks as raw disks
- ASM disks as Veritas Volume Manager volumes
- ASM disks as Veritas Cluster Volume Manager volumes

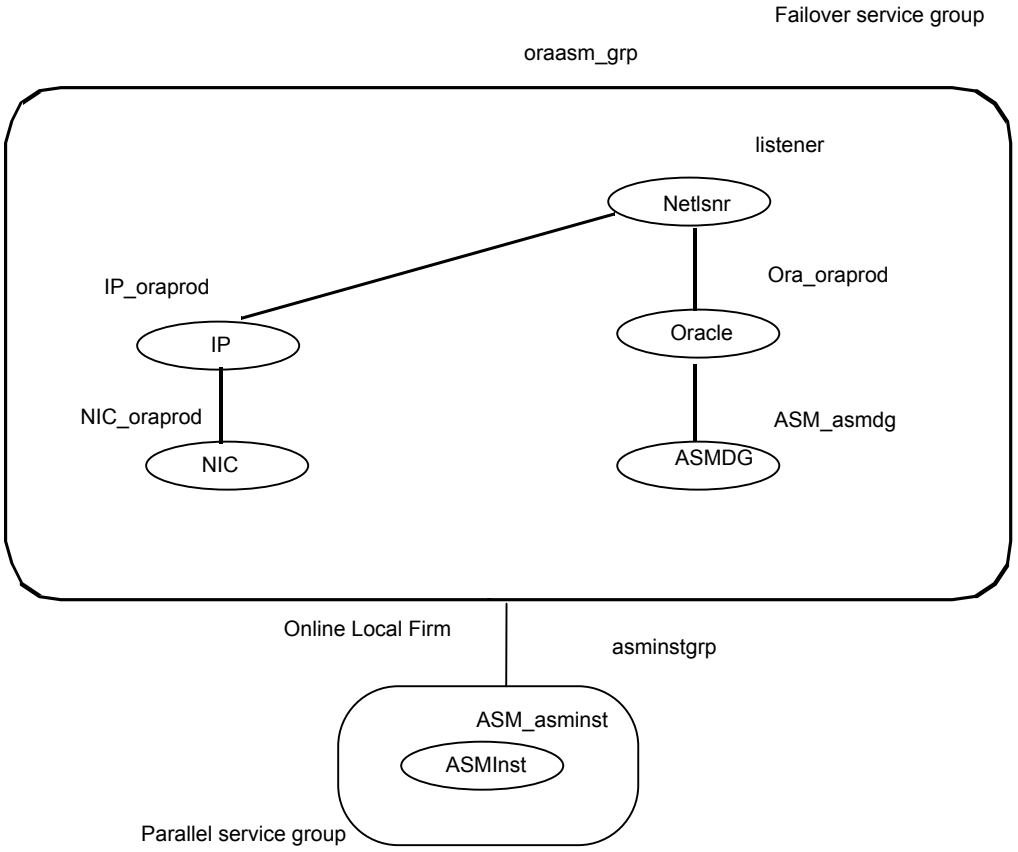
Sample configuration for ASM disks as raw disks

If you use raw devices as ASM disks, you can configure your service group in the following way:

- Oracle and ASMDG resources as parent failover service groups, and ASMInst resource as parallel service group
- The service groups are made dependent using an Online Local Firm dependency.

Figure B-17 describes a typical service group with ASMInst resource as a parallel service group.

Figure B-17 Dependency graph with ASMInst resource as a parallel service group



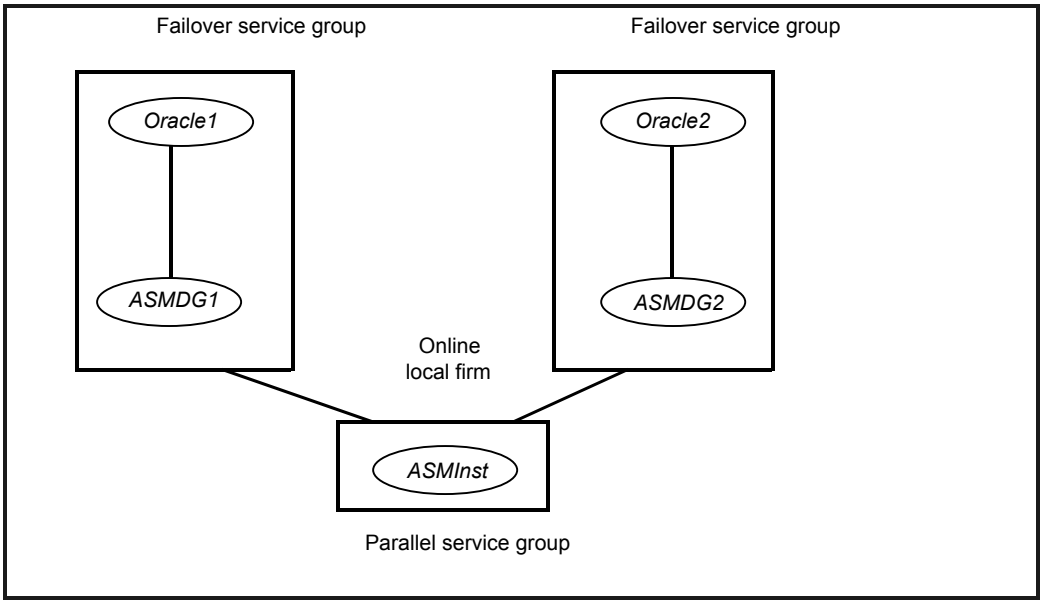
The Oracle ASM instance and ASM disk groups in the cluster are configured as resources of type ASMInst and ASMDG respectively. The ASMInst agent is configured as parallel service group, asminstgrp.

The virtual IP address for the service group is configured using the IP and NIC resource types. The Oracle and ASMDG resources are configured as failover service group, oraasm_grp. The Oracle server can be started after each of these resources is brought online.

The oraasm_grp is made dependent on the asminstgrp using an Online Local Firm dependency.

Figure B-18 describes a typical service group with multiple Oracle instances sharing an ASMInst resource that is configured as a parallel service group.

Figure B-18 Dependency graph for Oracle ASM with multiple Oracle instances on a node



If you have multiple Oracle instances, a failover service group is configured for each Oracle instance. The Oracle service groups share a single ASM instance that is configured as a parallel service group. The Oracle service groups are made dependent on the ASMInst service group using an Online Local Firm dependency. However, each database must use exclusive ASM disk groups, so that the Cluster Server agent for Oracle can fail over the disk group.

Sample VCS configuration file for ASM-managed Oracle instance

Review the sample configuration for an Oracle instance that is ASM-managed. The sample file has the ASMInst resource as part of a parallel service group.

```
include "types.cf"
include "OracleTypes.cf"
include "OracleASMTypes.cf"

cluster vcs (
)

system symnode01 (
```

```
)

system symnode02 (
)

group asminstgrp (
    SystemList = { symnode01 = 0, symnode02 = 1 }
    Parallel = 1
    )

    ASMInst ASM_asminst (
        Sid = "+ASM"
        Owner = "oraprod"
        Home = "/orahome/Oracle"
        )
group oraasm_grp (
    SystemList = { symnode01 = 0, symnode02 = 1 }
    AutoStartList = { symnode01 }
    )

    ASMDG ASM-asmdg (
        Sid = "+ASM"
        Owner = "oraprod"
        Home = "/orahome/Oracle"
        DiskGroups = { asmhighdg }
        )

    IP IP_oraprod (

        Device = hme0

        Address = "192.168.1.22"
        NetMask = "255.255.240.0"
    )

    NIC NIC_oraprod (

        Device = hme0


    NetworkHosts = { "192.168.1.1" }
    )

    Netlsnr LSNR_oraprod_lsnr (
```

```
    Owner = "oraprod"
    Home = "/orahome/Oracle"
    TnsAdmin = "/orahome/Oracle/network/admin"
    Listener = LISTENER_PROD
    MonScript = "./bin/Netlsnr/LsnrTest.pl"
)

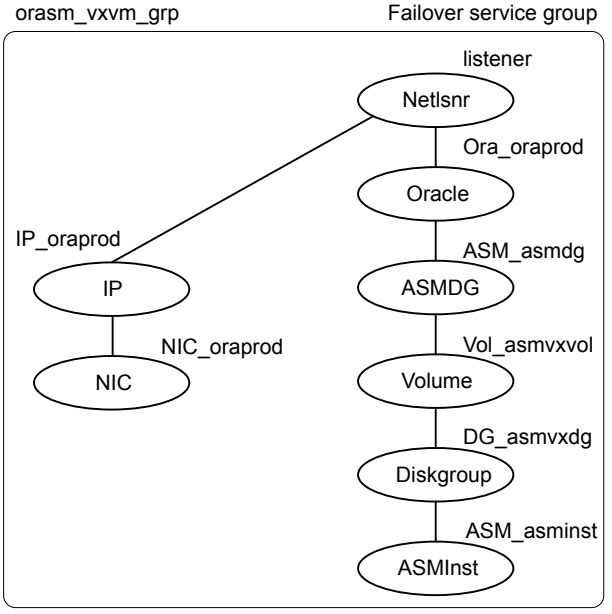
Oracle ORA_oraprod (
    Sid = PROD
    Owner = "oraprod"
    Home = "/orahome/Oracle"
    EnvFile = "/tmp/env.sh"
    MonScript = "./bin/Oracle/SqlTest.pl"
    User = thor
    Pword = hvlTptWvj
    Table = thor
    MonitorOption = 0
)

requires group asminstgrp online local firm
IP_oraprod requires NIC_oraprod
LSNR_oraprod_lsnr requires IP_oraprod
LSNR_oraprod_lsnr requires ORA_oraprod
ORA_oraprod requires ASM_asmdg
```

Sample configuration for ASM disks as VxVM volumes

Figure B-19 describes a typical service group with VxVM disks chosen for Oracle ASM.

Figure B-19 Dependency graph for ASM on VxVM disks

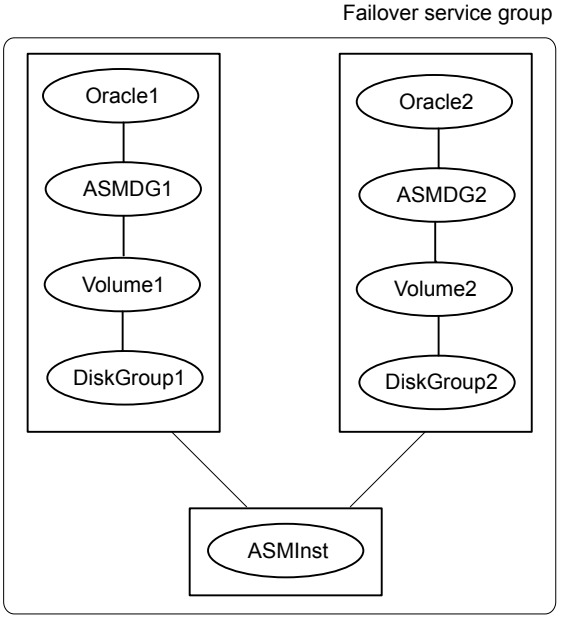


This configuration has a single failover service group. The Oracle ASM instance and ASM disk groups in the cluster are configured as resources of type ASMInst and ASMDG respectively.

The VxVM disk groups and volumes in the cluster are configured as resources of type DiskGroup and Volume respectively. The virtual IP address for the service group is configured using the IP and NIC resource types. The Oracle server can be started after each of these resources is brought online.

Figure B-20 describes a typical service group with multiple Oracle instances sharing the ASMInst resource that is part of a single failover service group.

Figure B-20 Dependency graph for Oracle ASM with multiple Oracle instances on a node



Sample VCS configuration file for ASM disks as VxVM disks

Review the sample configuration for an Oracle instance that is ASM-managed.

```
include "types.cf"
include "OracleTypes.cf"
include "OracleASMTypes.cf"

cluster vcs (
)

system symnode01 (
)

system symnode02 (
)

group orasm_vxvm_grp (
    SystemList = { symnode01 = 0, symnode02 = 1 }
    AutoStartList = { symnode01 }
```

```
)

    ASMDG ASM-asmdg (
        Sid = "+ASM"
        Owner = oraprod
        Home = "/orahome/Oracle"
        DiskGroups = { asmhighdg }

        DBAUser = “orauser”
        DBAPword = “DPSrFPeRIrGPiRF”

        )

    ASMInst ASM_asminst (
        Sid = "+ASM"
        Owner = oraprod
        Home = "/orahome/Oracle"

        DBAUser = “orauser”
        DBAPword = “DPSrFPeRIrGPiRF”

        )

    DiskGroup DG_asmvxdg (
        DiskGroup = asmvxdg

)

    IP IP_oraprod (

        Device = hme0

        Address = "192.168.1.22"
        NetMask = "255.255.240.0"
    )

    NIC NIC_oraprod (

        Device = hme0


   NetworkHosts = { "192.168.1.1" }
   )

    Netlsnr LSNR_oraprod_lsnr (
        Owner = oraprod
```

```
    Home = "/orahome/Oracle"
    TnsAdmin = "/orahome/Oracle/network/admin"
    Listener = LISTENER_PROD
    MonScript = "./bin/Netlsnr/LsnrTest.pl"
    LsnrPwd = cqfOdoOolOo
)

Oracle ORA_oraprod (
    Sid = PROD
    Owner = oraprod
    Home = "/orahome/Oracle"
    EnvFile = "/tmp/env.sh"
    MonScript = "./bin/Oracle/SqlTest.pl"
    User = thor
    Pword = hvlTptWvj
    Table = thor
    MonitorOption = 0
)

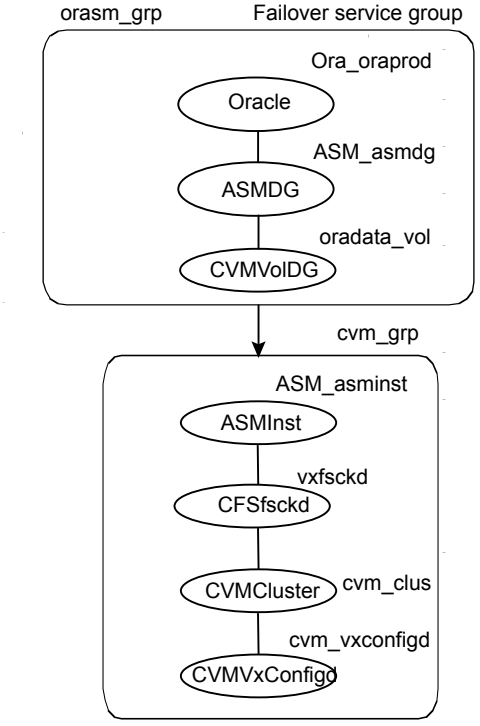
Volume Vol_asmvxvol (
    Volume = asmvxvol
    DiskGroup = asmvxdg
)

IP_oraprod requires NIC_oraprod
LSNR_oraprod_lsnr requires IP_oraprod
LSNR_oraprod_lsnr requires ORA_oraprod
ASM_asmdg requires VOL_asmvxvol
ORA_oraprod requires ASM_asmdg
VOL_asmvxvol requires DG_asmvxdg
DG_asmvxdg requires ASM_asminst
```

Sample configuration for ASM disks as CVM volumes

Figure B-21 describes a typical service group with CVM volumes chosen for Oracle ASM.

Figure B-21 Dependency graph for ASM on CVM volumes

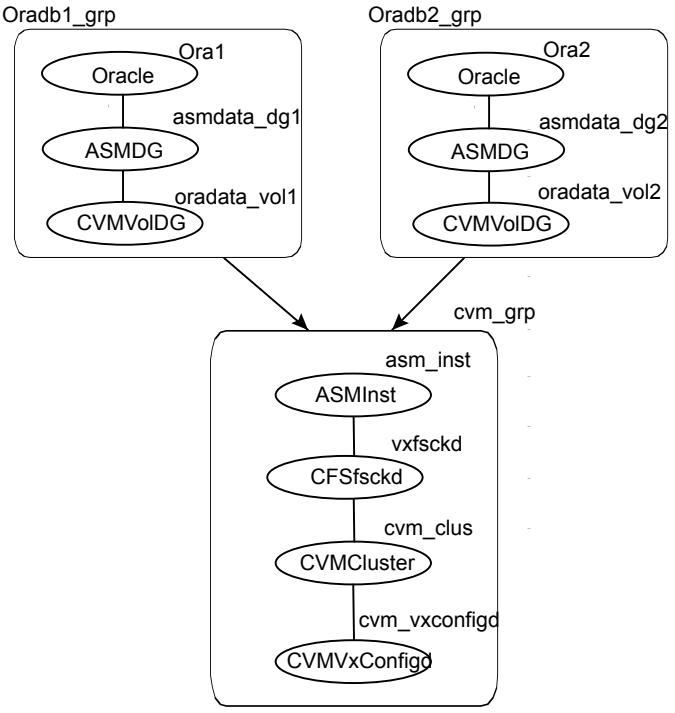


This configuration has two service groups. The Oracle, ASMDG, and CVMVolDG resources are part of the parent failover service group oraasm_grp. The ASMInst resource belongs to the CVM service group cvm_grp, which is a parallel service group. The service groups are linked with online local firm dependency.

After the CVM volume where the database resides comes online, the ASMDG agent mounts the ASM disk group that the database requires. The virtual IP address for the service group is configured using the IP and NIC resource types. The Oracle server can be started after each of these resources come online.

Figure B-22 describes a typical service group with multiple Oracle instances which share the ASMInst resource that is part of a CVM parallel service group.

Figure B-22 Dependency graph for Oracle ASM with multiple Oracle instances on a node



If you have multiple Oracle instances, a failover service group is configured for each Oracle instance. The Oracle service groups share a single ASM instance that is configured as part of the CVM parallel service group. The Oracle service groups are made dependent on the CVM service group using an Online Local Firm dependency. However, each database must use exclusive ASM disk groups, so that the Cluster Server agent for Oracle can fail over the disk group.

Sample VCS configuration file for ASM that uses CVM volumes

Review the sample configuration for an Oracle instance that uses CVM volumes for ASM.

```
include "types.cf"
include "CFSTypes.cf"
include "CVMTypes.cf"
include "OracleTypes.cf"
include "OracleASMTypes.cf"
```

```
cluster vcsclus_asm (
 UserNames = { admin = abcdef }
 Administrator = { admin }
 UseFence = SCSI3
 HacliUserLevel = COMMANDROOT
 )

system symnode01 (
 )

system symnode02 (
 )


group cvm (
 SystemList = { symnode01 = 0, symnode02 = 1 }
 AutoFailOver = 0
 Parallel = 1
 AutoStartList = { symnode01, symnode02 }
 )

 CFSfsckd vxfsckd (
  )

 CVMCluster cvm_clus (
  CVMClustName = vcsclus_asm
  CVMNodeId = { symnode01 = 0, symnode02 = 1 }
  CVMTransport = gab
  CVMTimeout = 200
  )

 CVMVxconfigd cvm_vxconfigd (
  Critical = 0
  CVMVxconfigdArgs = { syslog }
  )

 ASMInst ASM_asminst (
  Sid = "+ASM"
  Owner = oracle
  Home = "/orahome/Oracle"
  )
```

```
 ASM_asminst requires vxfsckd
 vxfsckd requires cvm_clus
 cvm_clus requires cvm_vxconfigd


group oraasm_grp (
 SystemList = { symnode01 = 0, symnode02 = 1 }
 AutoFailOver = 1
 AutoStartList = { symnode01, symnode02 }
 )

 CVMVolDg oradata_vol (
  CVMDiskGroup = ora_dg
  CVMVolume = { oradatavol }
  CVMActivation = sw
  )

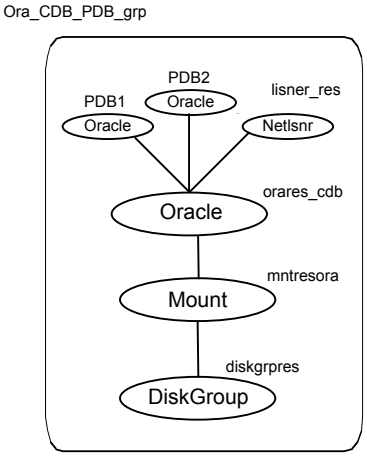
 ASMDG ASM_asmdg (
  Sid = "+ASM"
  Owner = oracle
  Home = "/orahome/Oracle"
  DiskGroups = { ASM_DG }
  )

 Oracle Ora_oraprod (
  Owner = oracle
  Sid = vrts
  Home = "/orahome/Oracle"
  )

 requires group cvm_grp online local firm
 Ora_oraprod requires ASM_asmdg
 ASM_asmdg requires oradata_vol
```

Sample configuration of Oracle pluggable database (PDB) resource in main.cf

Figure B-23 Sample Oracle CDB-PDB configuration



Review the sample configuration with two PDB resources that are configured in the main.cf file.

```
group CDB (
     SystemList = { sys1 = 1, sys2 = 1 }
     AutoStartList = { sys2 }
     )

     DiskGroup diskgrpres (
             DiskGroup = cdb_dg
             )

     Mount mntresora (
             MountPoint = "/db01"
             BlockDevice = "/dev/vx/dsk/cdb_dg/oravol"
```

```
        )

IP IP_oraprod (
        Device = eth0
        Address = "192.168.1.22"
        NetMask = "255.255.240.0"
        )

NIC NIC_oraprod (
        Device = eth0
         NetworkHosts = {"192.168.1.1"}
        )

Netlsnr lisner_res(
        Owner = oracle
        Home = "/u01/orahome"
        TnsAdmin = "/u01/orahome"
        Listener = LISTENER_PROD
        MonScript = "./bin/Netlsnr/LsnrTest.pl"
        LsnrPwd = cqfOdoOolOo
        )

Oracle orares_cdb (
        Critical = 0
        Sid = cdb
        Owner = oracle
        Home = "/u01/oraHome"
        StartUpOpt = STARTUP
        DBName = cdb
        User = "c##vcsuser"
        Pword = hvnTkvKvmVktMvj
        Table = vcs
        LevelTwoMonitorFreq = 5
        )


Oracle pdb1 (
        Critical = 0
        Sid = cdb
        Owner = oracle
        Home = "/u01/oraHome"
        StartUpOpt = STARTUP
        PDBName = pdb1
```

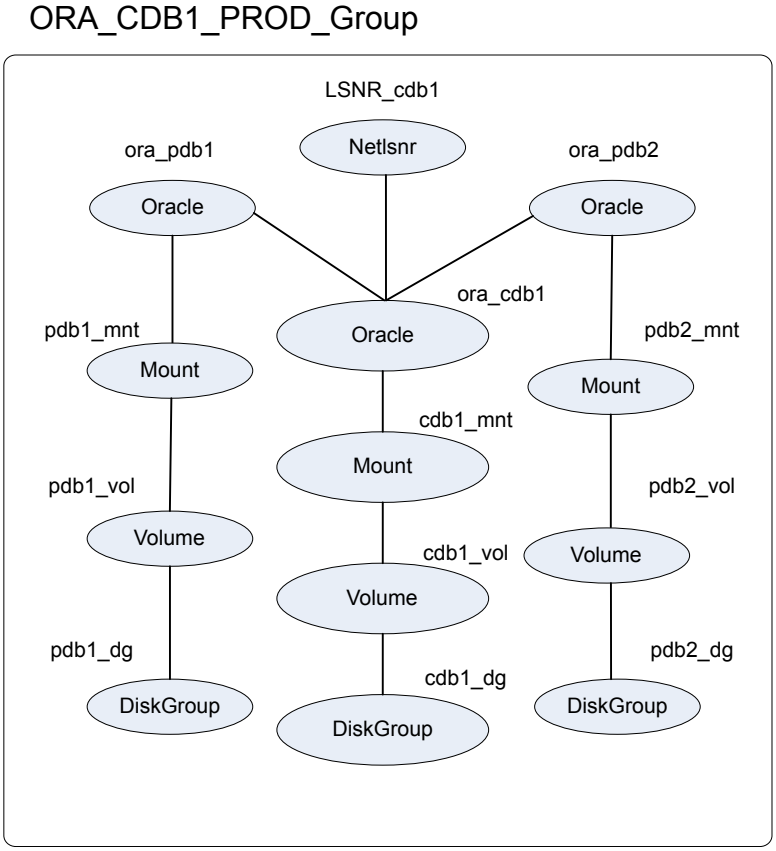
```
        User = "c##vcsuser"
        Pword = hvnTkvKvmVktMvj
        Table = vcs
        LevelTwoMonitorFreq = 5
        )

Oracle pdb2 (
        Critical = 1
        Sid = cdb
        Owner = oracle
        Home = "/u01/oraHome"
        PDBName = PDB2
        User = "c##vcsuser"
        Pword = hvnTkvKvmVktMvj
        Table = vcs
        LevelTwoMonitorFreq = 5
        )

mntresora requires diskgrpres
orares_cdb requires mntresora
pdb1 requires orares_cdb
pdb2 requires orares_cdb
lisner_res requires orares_cdb
IP_oraprod requires NIC_oraprod
lisner_res requires IP_oraprod
```

Sample configuration of migratable Oracle pluggable database (PDB) resource in main.cf

Figure B-24 Sample migratable Oracle CDB-PDB configuration




```
include "types.cf"
include "OracleTypes.cf"

cluster vcs (
)

system SysA (
)

system SysB(
```

```
)

group ORA_CDB1_PROD_Group(
 SystemList = { SysA = 0, SysB = 1 }
 )

        DiskGroup cdb1_dg (
                DiskGroup = cdb_dg1
                )

 Volume cdb1_vol (
  Volume = cdbvol1
  DiskGroup = cdb_dg1
  )

 Mount cdb1_mnt (
  MountPoint = "/cdb1"
  BlockDevice = "/dev/vx/dsk/cdb_dg1/cdbvol1"
  FSType = vxfs
  FsckOpt = "-y"
   )

 DiskGroup pdb1_dg (
                DiskGroup = pdb_dg1
                )

        Volume pdb1_vol (
                Volume = pdbvol1
                DiskGroup = pdb_dg1
                )

        Mount pdb1_mnt (
                MountPoint = "/pdb1"
                BlockDevice = "/dev/vx/dsk/pdb_dg1/pdbvol1"
                FSType = vxfs
                FsckOpt = "-y"
                )

 DiskGroup pdb2_dg (
                DiskGroup = pdb_dg2
                )

        Volume pdb2_vol (
```

```
                Volume = pdbvol2
                DiskGroup = pdb_dg2
                )

        Mount pdb2_mnt (
                MountPoint = "/pdb2"
                BlockDevice = "/dev/vx/dsk/pdb_dg2/pdbvol2"
                FSType = vxfs
                FsckOpt = "-y"
                )

IP IP_cdb1 (
                Device = hme0
                Address = "10.212.102.13"
                NetMask = "255.255.252.0"
                )

NIC NIC_cdb1 (
                Device = hme0
                NetworkHosts = { "10.212.102.1" }
                )

Netlsnr LSNR_cdb1 (
                Owner = oracle
                Home = "/u01/oraHome1"
                )

Oracle ora_cdb1 (
 Sid = cdb1
 Owner = oracle
 Home = "/u01/oraHome1"
 StartUpOpt = STARTUP
 DBName = cdb1
 )

Oracle ora_pdb1 (
 Sid = cdb1
 Owner = oracle
 Home = "/u01/oraHome1"
 PDBName = PDB1
 )

Oracle ora_pdb2 (
```

```
    Critical = 0
    Sid = cdb1
    Owner = oracle
    Home = "/u01/oraHome1"
    PDBName = PDB2
    )

    IP_cdb1 requires NIC_cdb1
    LSNR_cdb1 requires IP_cdb1
    LSNR_cdb1 requires ora_cdb1
    cdb1_mnt requires cdb1_vol
    cdb1_vol requires cdb1_dg
    pdb1_mnt requires pdb1_vol
    pdb1_vol requires pdb1_dg
    pdb2_mnt requires pdb2_vol
    pdb2_vol requires pdb2_dg
    ora_cdb1 requires cdb1_mnt
    ora_pdb1 requires ora_cdb1
    ora_pdb1 requires pdb1_mnt
    ora_pdb2 requires ora_cdb1
    ora_pdb2 requires pdb2_mnt

   group ORA_CDB2_PROD_Group(
    SystemList = { SysA = 0, SysB = 1 }
    )

           DiskGroup cdb2_dg (
                   DiskGroup = cdb_dg2
                   )

    Volume cdb2_vol (
     Volume = cdbvol2
     DiskGroup = cdb_dg2
     )

    Mount cdb2_mnt (
     MountPoint = "/cdb2"
     BlockDevice = "/dev/vx/dsk/cdb_dg2/cdbvol2"
     FSType = vxfs
     FsckOpt = "-y"
     )

    DiskGroup pdb3_dg (
```

```
                DiskGroup = pdb_dg3
                )

        Volume pdb3_vol (
                Volume = pdbvol3
                DiskGroup = pdb_dg3
                )

        Mount pdb3_mnt (
                MountPoint = "/pdb3"
                BlockDevice = "/dev/vx/dsk/pdb_dg3/pdbvol3"
                FSType = vxfs
                FsckOpt = "-y"
                )

DiskGroup pdb4_dg (
                DiskGroup = pdb_dg4
                )

        Volume pdb4_vol (
                Volume = pdbvol4
                DiskGroup = pdb_dg4
                )

        Mount pdb4_mnt (
                MountPoint = "/pdb4"
                BlockDevice = "/dev/vx/dsk/pdb_dg4/pdbvol4"
                FSType = vxfs
                FsckOpt = "-y"
                )

IP IP_cdb2 (
                Device = hme0
                Address = "10.212.102.14"
                NetMask = "255.255.252.0"
                )

NIC NIC_cdb2 (
                Device = hme0
                NetworkHosts = { "10.212.102.1" }
                )

Netlsnr LSNR_cdb2 (
```

```
                Owner = oracle
                Home = "/u01/oraHome2"
                )

Oracle ora_cdb2 (
 Sid = cdb2
 Owner = oracle
 Home = "/u01/oraHome2"
 StartUpOpt = STARTUP
 DBName = cdb2
 )

Oracle ora_pdb3 (
 Sid = cdb2
 Owner = oracle
 Home = "/u01/oraHome2"
 PDBName = PDB3
 )

Oracle ora_pdb4 (
 Sid = cdb2
 Owner = oracle
 Home = "/u01/oraHome2"
 PDBName = PDB4
 )

IP_cdb2 requires NIC_cdb2
LSNR_cdb2 requires IP_cdb2
LSNR_cdb2 requires ora_cdb2
cdb2_mnt requires cdb2_vol
cdb2_vol requires cdb2_dg
pdb3_mnt requires pdb3_vol
pdb3_vol requires pdb3_dg
pdb4_mnt requires pdb4_vol
pdb4_vol requires pdb4_dg
ora_cdb2 requires cdb2_mnt
ora_pdb3 requires ora_cdb2
ora_pdb3 requires pdb3_mnt
ora_pdb4 requires ora_cdb2
ora_pdb4 requires pdb4_mnt
```

Sample Configuration of Oracle supported by systemD

```
include "types.cf"
include "OracleTypes.cf"

cluster vcs (
        )

system sys1 (
        )

system sys2 (
        )

group orasg1 (
        SystemList = { sys1 = 0, sys2 = 1 }
        )

DiskGroup DG_oraprod (
Critical = 0
DiskGroup = oradb_dg
                )
IP IP_oraprod (
Critical = 0
Device @sys1 = eth0
Device @sys2 = eno16777728
Address = "123.123.145.42"
NetMask = "255.255.255.0"
                )

Mount Mount_oraprod (
Critical = 0
MountPoint = "/home/oracle/oradata"
BlockDevice = "/dev/vx/dsk/oradb_dg/oradb_vol"
FSType = vxfs
MountOpt = rw
FsckOpt = "-y"
         )
NIC NIC_Public1 (
Critical = 0
```

```
Device @sys1 = eth0
Device @sys2 = eno16777728
NetworkHosts = { "123.123.123.1" }
               )

Netlsnr LSNR_oracle (
Critical = 0
Owner = oracle
Home = "/home/oracle/app/oracle/product/dbhome_1"
TnsAdmin = "/home/oracle/app/oracle/product/dbhome_1/network/admin"
EnvFile = "/home/oracle/.bashrc"
MonScript = "/opt/VRTSagents/ha/bin/Netlsnr/LsnrTest.pl"
               )

Oracle ORA_oraprod (
SystemDAttrList = { LimitLOCKS = 44432 }
Sid = oradb
Owner = ORA_oraprod
Home = "/home/oracle/app/oracle/product/dbhome_1"
Pfile = "/home/oracle/app/oracle/product/dbhome_1/dbs/initoradb.ora"
StartUpOpt = STARTUP
EnvFile = "/tmp/a"
UseSystemD = 1
MonScript = "/opt/VRTSagents/ha/bin/Oracle/SqlTest.pl"
User = vcsuser
Pword = aogMdoDofOdmFoc
Table = vcs
LevelTwoMonitorFreq = 0
               )

Volume Vol_oraprod (
Critical = 0
DiskGroup = oradb_dg
Volume = oradb_vol
               )

IP_oraprod requires NIC_Public1
LSNR_oracle requires IP_oraprod
Mount_oraprod requires Vol_oraprod
ORA_oraprod requires LSNR_oracle
ORA_oraprod requires Mount_oraprod
Vol_oraprod requires DG_oraprod
```

Sample configuration of ASMInst supported by systemD

```
ASMInst Sample main.cf
include "OracleASMTypes.cf"
include "types.cf"
include "OracleTypes.cf"

cluster vcs (
        UserNames = { admin = Hk, admini = gmnLmjMj }
        Administrators = { admini }
        )

system sys1 (
        )

system sys2 (
        )

group asminstgrp (
 SystemList = { sys1 = 0, sys2 = 1 }
 )

 ASMInst ASM_asminst (
                SystemDAttrList = { LimitNOFILE = 55588 }
                Sid = "+ASM"
                Owner = oracle
                Home = "/u01/app"
                StartUpOpt = STARTUP_MOUNT
                UseSystemD = 1
                )

group ora_asmgrp (
        SystemList = { sys1 = 0, sys2 = 1 }
        )

        ASMDG ASM_asmdg(
                Sid = "+ASM"
                Owner = oracle
                Home = "/u01/app"
                DiskGroups = { DBDG }
                )
```

```
Oracle Ora_oraprod (
        Sid = oradb
        Owner = oracle
        Home = "/home/oracle/oracle"
        Pfile = "/home/oracle/oracle/dbs/initoradb.ora"
        StartUpOpt = SRVCTLSTART
        ShutDownOpt = SRVCTLSTOP
        MonScript = "/opt/VRTSagents/ha/bin/Oracle/SqlTest.pl"
        )

   requires group asminstgrp online local firm
Ora_oraprod requires ASM_asmdg
```

Appendix C

Best practices

This appendix includes the following topics:

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

Best practices for multiple Oracle instance configurations in a VCS environment

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

- For each SID to be configured, create UNIX accounts with DBA privileges.
- Make sure that each Oracle instance has a separate disk group and is configured as a separate service group.
- Define the system parameters such that the allocation of semaphore and shared memory is appropriate on all systems.
- 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 to ensure clean failover.
- If your configuration has different versions of Oracle, create a separate $ORACLE_HOME for each Oracle version.
- Follow the Optimal Flexible Architecture (OFA) standard (/uxx/<SID>). In cluster configurations, you could adapt the standard to make it more application-specific. For example, /app/uxx/<SID>.
- 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 cluster. 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.
- The pfiles must be coordinated between systems. For the same instance of a database, ensure that the pfiles referenced are identical across the nodes.

Appendix D

Using the SPFILE in a VCS cluster for Oracle

This appendix includes the following topics:

- About the Oracle initialization parameter files
- Starting an Oracle instance
- Using the SPFILE in a VCS cluster

About the Oracle initialization parameter files

Oracle9i introduced the SPFILE, which is a binary file stored on the database server. With this feature, changes to the instance parameters can be set to be persistent across all startup and shutdown procedures.

Starting an Oracle instance

You can start an Oracle instance in the following ways:

- Using the default SPFILE spfile*SID*.ora
- Using the default init.ora file init*SID*.ora
- By specifying an initialization file init.ora
- By specifying an SPFILE in the initialization file init.ora

When you run the `startup` command without a PFILE clause, Oracle reads the initialization parameters from the SPFILE. On Solaris platforms, the default location for the SPFILE or PFILE is $ORACLE_HOME/dbs.

Oracle locates the initialization parameter file by examining file names in the following order:

- SPFILE*SID*.ora
- SPFILE.ora
- init*SID*.ora

Using the SPFILE in a VCS cluster

Oracle versions earlier than Oracle9i used an initialization file initSID.ora, a text file, to start database instances. Changes that were applied to instance parameters during a database session were not saved to the file. You had to manually apply them to the initialization file.

When using the Cluster Server 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: If you want the SPFILE's session parameter changes be persistent across an instance failover, then Veritas recommends you to save the SPFILE on shared storage.

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 Solaris).

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 E

OHASD in a single instance database environment

This appendix includes the following topics:

- About OHASD in a single instance database environment
- Configuring the Application agent to make OHASD highly available

About OHASD in a single instance database environment

The Oracle Grid Infrastructure software provides system support for an Oracle database including Oracle Automatic Restart and Oracle ASM features.

On nodes that have Oracle Grid Infrastructure installed in a single instance database environment, ensure that the Oracle High Availability Services daemon (OHASD) process comes online before the Oracle resources.

Administration of Oracle resources on these nodes is dependent on the OHASD process. So, the process needs to be online first to ensure that the resources come online.

Configuring the Application agent to make OHASD highly available

You can configure the Application agent to ensure the OHASD process is made highly available in the VCS cluster.

To configure the Application agent for the Ohasd process to ensure the service is online

1. Configure the Application agent

```
Application ohasd_res (
   StartProgram = "<GRID_HOME>/bin/crsctl start has"
   StopProgram = "<GRID_HOME>/bin/crsctl stop has"
   PidFiles @node1 = {
            "<GRID_HOME>/ohasd/init/node1.pid" }
   PidFiles @node2 = {
            "<GRID_HOME>/ohasd/init/node2.pid" }
   )
```

2. Set up the resource dependency on the Ohasd daemon if the resource is configured:
   - For databases that are running in oracle restart mode
   - For the ASMInst resource
3. In the following dependency tree, ohasd resource is configured in a failover service group or parallel service group.

   Run the `# $GRID_HOME/bin/srvctl disable has` command to disable the automatic startup of ohasd on all the nodes in the VCS cluster.

Figure E-1 shows OHASD resource configured in Parallel service group with the ASMInst resource in the Failover service group

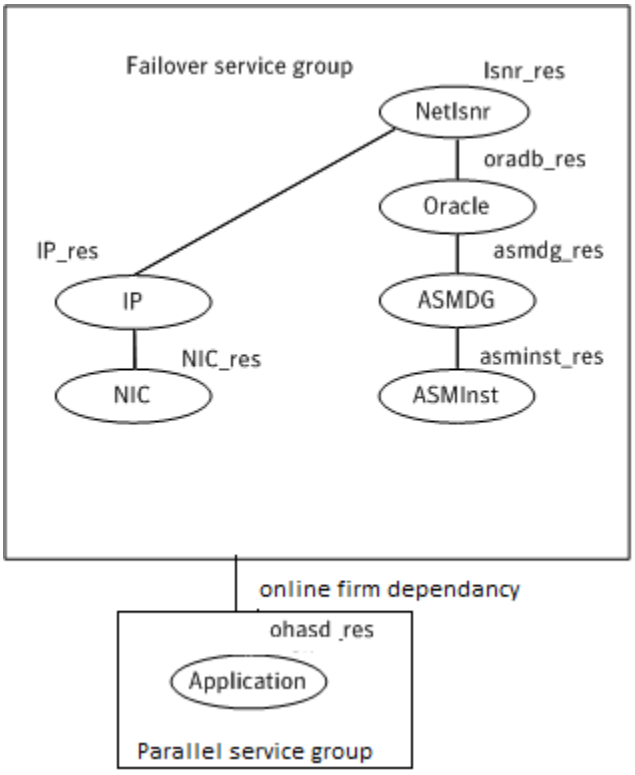


Figure E-2 shows OHASD resource configured in Failover service group with the ASMInst resource.

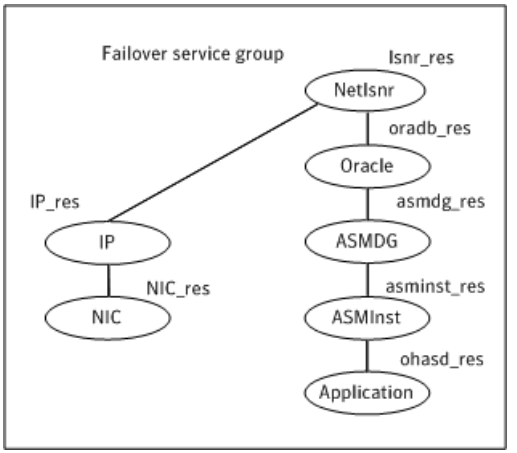


Figure E-3 shows the dependency tree of databases that are running in oracle restart mode.

