

VERITAS Cluster Server™ Enterprise Agent 4.0 for Oracle

Installation and Configuration Guide

Solaris

N10054F

January 2004

Disclaimer

The information contained in this publication is subject to change without notice. VERITAS Software Corporation makes no warranty of any kind with regard to this manual, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. VERITAS Software Corporation shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance, or use of this manual.

VERITAS Legal Notice

Copyright © 1998-2004 VERITAS Software Corporation. All rights reserved. VERITAS, VERITAS Software, the VERITAS logo, VERITAS Cluster Server, and all other VERITAS product names and slogans are trademarks or registered trademarks of VERITAS Software Corporation. VERITAS, the VERITAS Logo, and Cluster Server Reg. U.S. Pat. & Tm. Off. Other product names and/or slogans mentioned herein may be trademarks or registered trademarks of their respective companies.

VERITAS Software Corporation
350 Ellis Street
Mountain View, CA 94043
USA
Phone 650-527-8000 Fax 650-527-2901
www.veritas.com



Contents

Preface	vii
How This Guide is Organized	vii
Getting Help	viii
Telephone and Fax Support	viii
Conventions	viii
 Chapter 1. Introduction	1
About the VCS Enterprise Agent for Oracle	1
Oracle Agent	2
Netlsnr Agent	3
New Features and Updates	4
Startup and Shutdown Options	4
Handling Oracle Error Codes During Detail Monitoring	6
Handling Attribute of Type Password	7
Info Entry Point	7
Action Entry Point	8
Supported Software	9
How the Agent Makes Oracle Highly Available	9
Typical Oracle Configuration in a VCS Cluster	10
Setting Up an Oracle Cluster	11
 Chapter 2. Installing and Configuring Oracle	13
Prerequisites	13
Points to Note Before Installing Oracle	14



Location of the \$ORACLE_HOME	14
Multiple Oracle Instances (SIDs)	15
Location of Database Tablespaces	15
Configuring the Location of Core Files	15
Transparent Listener Failover	15
Long Pathname Limitation for \$ORACLE_HOME	16
Defining NLS Information	16
Hot Backup of Oracle Database and VCS	17
Installing Oracle	18
Overview of Tasks	19
Installing Oracle Binaries	21
Configuring the Database	25
Copying the \$ORACLE_BASE/admin/SID Directory	31
Starting the Database	31
Chapter 3. Installing the Oracle Agent	33
Installing the Agent Software	33
Prerequisites	33
Installation Instructions	33
Upgrading the Agent	35
Chapter 4. Configuring the Agent	37
Prerequisites	37
Oracle Configurations in VCS	38
Single Oracle Instance Configuration	39
Multiple Oracle Instances (Single Listener)	41
Multiple Oracle Instances (Multiple Listeners)	44
Oracle With Shared Server Support	46
Configuring the Agent Using the Agent Configuration Wizard	49
Configuring the Agent Using Cluster Manager (Java Console)	59
Importing the OracleTypes.cf File	59



Creating a New Service Group	60
Editing Resource attributes	61
Configuring the Agent Using the Command-Line	62
Encrypting Passwords	64
Additional Considerations with Basic Monitoring	65
Setting Up Detail Monitoring	65
Setting Up Detail Monitoring for Oracle	66
Enabling and Disabling Detail Monitoring for Oracle	68
Setting Up Detail Monitoring for Netlsnr	68
Chapter 5. Administering the Oracle Service Group	69
Bringing the Service Group Online	69
Switching the Service Group	70
Taking the Service Group Offline	70
Disabling the Agent	71
Modifying the Agent Configuration	72
Removing the Agent	72
Chapter 6. Troubleshooting	73
Error Messages Common to the Oracle and Netlsnr Agents	73
Error Messages Specific to the Oracle Agent	75
Error Messages Specific to the Netlsnr Agent	77
Appendix A. Resource Type Definitions	79
Oracle Resource Type	79
Attribute Definitions	80
Netlsnr Resource Type	83
Attribute Definitions	83
Appendix B. Sample Configurations	85
Single Instance Configuration	86

Multiple Oracle Instance (Single Listener) Configuration	89
Multiple Instance (Multiple Listeners) Configuration	95
Oracle Configuration With Shared Server Support	100
Appendix C. Best Practices for Multiple Oracle Instance Configurations	103
Appendix D. Using the SPFILE in an Oracle Cluster	105
Starting an Oracle Instance	105
Using the SPFILE in a VCS Cluster	106
Specifying the SPFILE Location in the PFILE	106
Creating the SPFILE from a PFILE	106
Index	107



Preface

This book describes how to install and configure the VERITAS Cluster Server™ (VCS) Enterprise Agent 4.0 for Oracle.

If this document is dated more than six months prior to the date you are installing your enterprise agent, contact VERITAS Technical Support to confirm you have the latest supported versions of the application and operating system.

How This Guide is Organized

Chapter 1. “Introduction” on page 1 presents a brief introduction of the VCS enterprise agent for Oracle.

Chapter 2. “Installing and Configuring Oracle” on page 13 describes how to install and configure Oracle in a VCS cluster.

Chapter 3. “Installing the Oracle Agent” on page 33 describes how to install VCS enterprise agent for Oracle in a VCS cluster.

Chapter 4. “Configuring the Agent” on page 37 describes how to configure an Oracle service group in a VCS cluster.

Chapter 5. “Administering the Oracle Service Group” on page 69 describes the administrative tasks you can perform on an Oracle service group.

Chapter 6. “Troubleshooting” on page 73 lists the error messages associated with the VCS enterprise agent for Oracle. If applicable, a recommended solution is also provided.

Appendix A. “Resource Type Definitions” on page 79 lists the resource type definitions and attribute definitions for the VCS enterprise agent for Oracle.

Appendix B. “Sample Configurations” on page 85 provides sample configurations for the Oracle service group.

Appendix C. “Best Practices for Multiple Oracle Instance Configurations” on page 103 lists some best practices for using the VCS enterprise agent for Oracle.

Appendix D. “Using the SPFILE in an Oracle Cluster” on page 105 describes how to use the SPFILE in an Oracle cluster.



Getting Help

For technical assistance, visit the VERITAS Technical Services Web site at <http://support.veritas.com>. From there you can:

- ◆ Contact the VERITAS Technical Services staff and post questions to them.
- ◆ Download the latest patches and utilities.
- ◆ View the VERITAS Cluster Server Frequently Asked Questions (FAQ) page.
- ◆ Search the knowledge base for answers to technical support questions.
- ◆ Receive automatic notice of product updates.
- ◆ Learn about VERITAS Cluster Server training.
- ◆ Read white papers related to VERITAS Cluster Server.
- ◆ Access the latest product documentation and technical notes.

Telephone and Fax Support

Telephone and fax support for VERITAS Cluster Server is available only with a valid support contract. To contact Technical Services, dial the appropriate phone number listed on the *Technical Support Guide* included in the product box. Have your product license information ready to ensure you are routed to the proper support personnel.

Conventions

Typeface/Font	Usage
bold	names of screens, windows, tabs, dialog boxes, options, buttons
<i>italic</i>	new terms, book titles, emphasis, variables in tables or body text
Courier	computer output, command references within text
Courier (bold)	command-line user input, keywords in grammar syntax
<i>Courier</i> (bold, italic)	variables in a command
Symbol	Usage
#	superuser prompt (for all shells)

Introduction

1

VCS enterprise agents monitor specific resources within an enterprise application, determine the status of these resources, and start or stop them according to external events. The VCS enterprise agent for Oracle provides high availability for Oracle in a VCS cluster.

About the VCS Enterprise Agent for Oracle

The VCS enterprise agent for Oracle monitors the Oracle and listener processes, brings them online, and takes them offline.

The package contains two VCS agents:

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

The agents include VCS type declarations and agent executables, and are represented with the Oracle and Netlsnr resource types, respectively. Both agents work together to make Oracle highly available in a VCS cluster.



Oracle Agent

The Oracle agent monitors the database processes. Specific agent operations include:

- ◆ **Online**—Starts the Oracle database by using the following `svrmgrl` or `sqlplus` command:

```
startup pfile=$PFile force
```

You can also configure the agent to start the database using different Startup options for Oracle. See “[Startup Options](#)” on page 4 for more information about these options.

- ◆ **Offline**—Stops the Oracle database with the specified options by using the following `svrmgrl` or `sqlplus` command:

```
shutdown immediate
```

You can also configure the agent to stop the database using different Shutdown options for Oracle. See “[Shutdown Options](#)” on page 5 for more information about these options.

- ◆ **Monitor**—The Oracle agent provides two levels of monitoring: *basic* and *detail*. In the basic monitoring mode, the agent scans the process table for the `ora_dbw`, `ora_smon`, `ora_pmon`, and `ora_lgwr` processes to verify Oracle is running. In the detail monitoring mode, the agent performs a transaction on a test table in the database to ensure that Oracle is functioning properly.
- ◆ **Clean**—Forcibly stops the Oracle database by using the following `svrmgrl` or `sqlplus` command:

```
shutdown abort
```

If the process does not respond to the `shutdown` command, the agent scans the process table for processes associated with the configured instance and kills them.

- ◆ **Info**—Provides static and dynamic information about the state of the database. See “[Info Entry Point](#)” on page 7 for more information.
- ◆ **Action**—Performs predefined actions on a resource. See “[Action Entry Point](#)” on page 8 for more information.

Netlsnr Agent

The Netlsnr agent monitors the listener process. The agent controls the listener for the database. 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. Specific agent operations include:

- ◆ Online—Starts the listener process by using the following command:

```
lsnrctl start $LISTENER
```

- ◆ Offline—Stops the listener process by using the following command:

```
lsnrctl stop $LISTENER
```

If the listener is configured with a password, the agent uses the password to stop the listener.

- ◆ Monitor—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 the listener process is running. In the detail monitoring mode, the agent uses the command `lsnrctl status $LISTENER` to verify the status of the Listener process.
- ◆ Clean—Scans the process table for `tnslsnr $Listener` and kills it.
- ◆ Action—Performs predefined actions on a resource. See “[Action Entry Point](#)” on page 8 for more information.



New Features and Updates

This section describes the new features and updates introduced in this version of the agent.

Startup and Shutdown Options

You can now specify Startup and Shutdown options for configured Oracle instances.

Startup Options

The agent supports the following startup options:

Startup Option	Description
STARTUP (Default)	Runs <code>startup pfile='location_of_pfile' force</code> if the pfile is configured. If the pfile is not configured, the agent runs <code>startup force</code> . It picks up the default parameter files from their default locations.
RESTRICTED	Starts the database in the RESTRICTED mode.
RECOVERDB	Performs a database recovery on instance startup.
CUSTOM	Uses a predefined SQL script (<code>start-custom-\$SID.sql</code>) and runs custom startup options. The script must be in the <code>\$VCS_HOME/bin/Oracle</code> 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: <pre>sqlplus /nolog <<! connect / as sysdba; @start_custom_\$SID.sql exit; !</pre>
SRVCTLSTART	Uses the <code>srvctl</code> utility to start an instance of the database. Use this option for RAC clusters for Oracle 9i R2 and later.



Shutdown Options

The agent supports the following shutdown options:

Shutdown Option	Description
IMMEDIATE (Default)	Shuts down the Oracle instance by running <code>shutdown immediate</code> .
SRVCTLSTOP	Sets up <code>srvctl</code> as the client tool and uses <code>srvctl</code> syntax to shut down the instance. This option is valid only for RAC instances in versions 9.2.0 and above.
TRANSACTIONAL	Runs the <code>shutdown transactional</code> command. This option is valid only for databases that support this option.
CUSTOM	Uses a predefined SQL script (<code>stop-custom-$\\$SID.sql</code>) and runs custom shutdown options. The script must be in the <code>$\\$VCS_HOME/bin/Oracle</code> 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 <code>srvctl</code> utility to stop an instance of the database. Use this option for RAC clusters for Oracle 9i R2 and later.



Handling Oracle Error Codes During Detail Monitoring

This release of the agent comes with enhanced handling of 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 called `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
```

The agent supports the following actions:

Action	Description
IGNORE	Ignores the error.
UNKNOWN	Marks the resource state as UNKNOWN and sends a notification if the Notifier resource is configured. See the <i>VERITAS Cluster Server User's Guide</i> 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, which fails over to the next available system. This is agent's default behavior. If the file <code>oraerror.dat</code> is not available, the agent assumes this default behavior for every Oracle error encountered.
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 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 will not help.



Handling Attribute of Type Password

User passwords used for detail monitoring of the Oracle database are no longer 'clear-text'. The Oracle agent supports encryption of user passwords. See [“Encrypting Passwords”](#) on page 64 for further information.

Configuring 'listener' passwords is introduced in the Netlsnr agent. This password is considered for querying the status of the listener and stopping the listener. This password is also encrypted. See [“Encrypting Passwords”](#) on page 64 for further information.

Info Entry Point

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

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 |
| ♦ TotalDBfileSize | ♦ FreeDBfileSize | ♦ Shared Pool Percent free |
| ♦ Buffer Hits Percent | | |

You can add additional attributes by adding `sql` statements to the file `$VCS_HOME/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

The Oracle agent 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>]
```

The agent supports the following 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 monitored by the agent. 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. This option is available for Oracle versions 8i and above.
DBResume	Resumes a suspended database. This option is available for Oracle versions 8i and above.
DBTbspBackup	Backs up a tablespace; <code>actionargs</code> contains name of the tablespace to be backed up.

You can also add custom actions for the agent. For further information, refer to the Agent Developer’s Guide.



Supported Software

- ◆ Oracle 8.0.x, Oracle 8.1.x, Oracle 9i, and Oracle 10g, including 64-bit versions
- ◆ VCS 4.0 on Solaris
- ◆ Solaris 2.6, 7, 8, and 9

How the Agent Makes Oracle Highly Available

The VCS enterprise agent for Oracle continuously monitors the Oracle and listener processes to verify they are functioning properly. The agent provides two levels of application monitoring:

- ◆ **Primary or Basic Monitoring**

In the Basic monitoring mode, the agent verifies that the Oracle and listener processes are present in the process table.

Note Basic monitoring cannot detect whether processes are in a hung or stopped states.

- ◆ **Secondary or Detail Monitoring**

In the Detail monitoring mode, the agent runs a perl script that executes commands against the database and listener to verify their status.

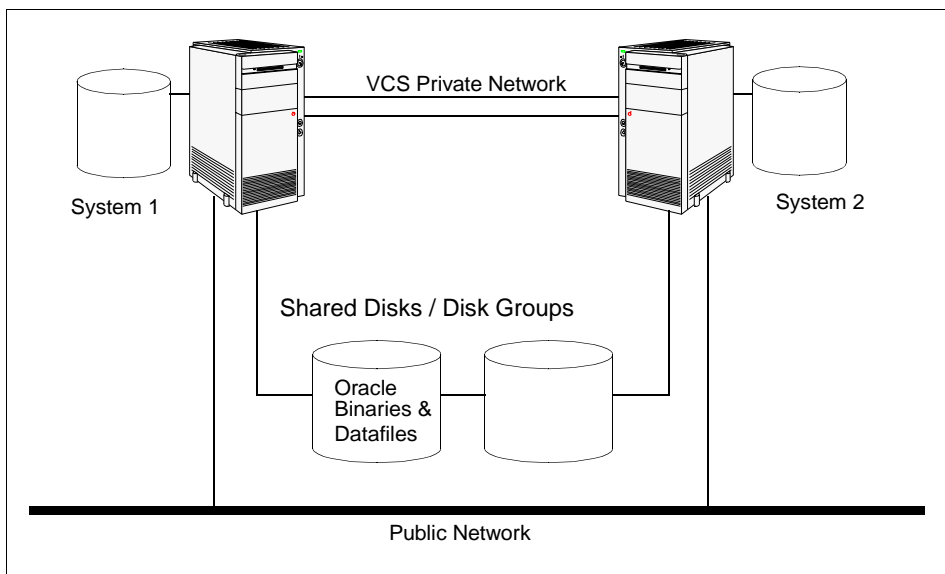
The agent detects application failure if the monitoring routine reports that the Oracle or listener processes are not functioning properly. When this occurs, the Oracle service group fails over to another node in the cluster, thus ensuring high availability for the Oracle services and the database.



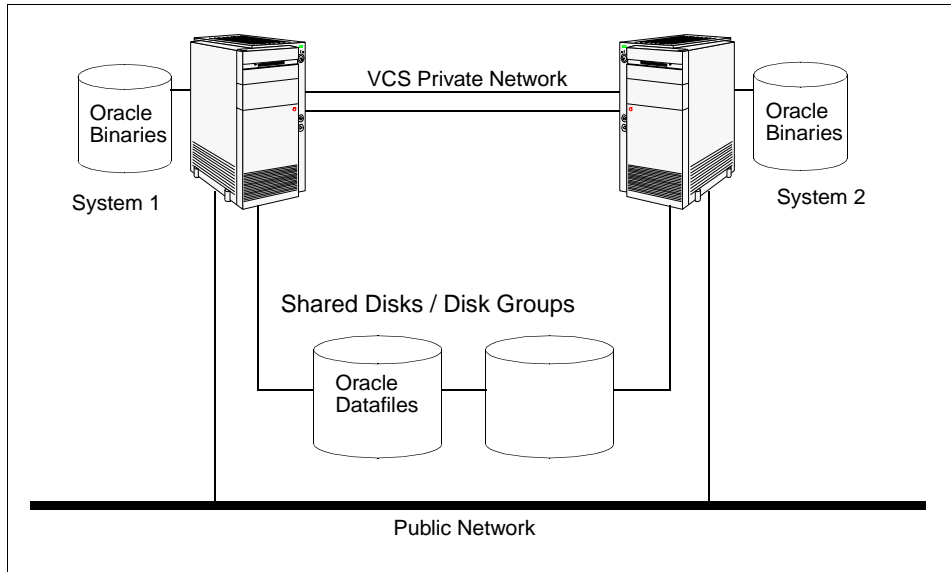
Typical Oracle Configuration in a VCS Cluster

In a typical configuration, VCS is configured in a two-system cluster (System 1 and System 2). The data to be used by Oracle is installed on shared storage. The Oracle binaries can be installed locally on both nodes or on shared disks. The Oracle enterprise agent is installed on both systems.

In the configuration depicted in the illustration below, Oracle binaries and data is installed completely on shared disks.

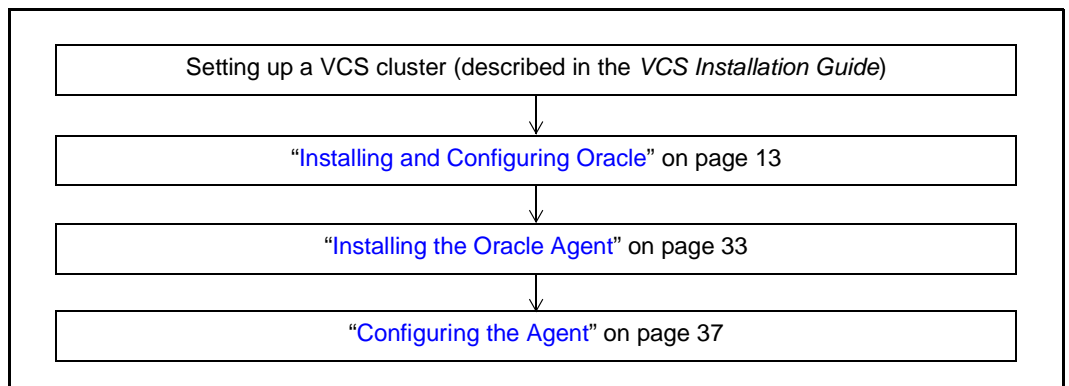


In the configuration depicted in the illustration below, Oracle binaries are installed locally on each node and Oracle data is installed on shared disks.



Setting Up an Oracle Cluster

Setting up an Oracle cluster in a VCS environment involves the following tasks. Each task is described in detail in subsequent chapters.





Installing and Configuring Oracle

2

This chapter describes how to install and configure Oracle and its components in a VCS cluster. The chapter also lists the prerequisites for installing Oracle. For more information on installing Oracle, refer to the Oracle documentation on Solaris.

Prerequisites

- ✓ Verify that all systems in the cluster have adequate resources to run Oracle and VCS.
- ✓ Verify that the network supports the TCP/IP protocol.



Points to Note Before Installing Oracle

The following sections discuss some important tasks that need to be noted before you install Oracle in a VCS cluster on Solaris.

Location of the \$ORACLE_HOME

You can install 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

When the Oracle Database Server (\$ORACLE_HOME) is installed on shared disks, each cluster system must have the same mount point directory for the shared file system. Placing Oracle binaries on shared storage simplifies setting up a given system in a cluster to run an instance. Each database service group is self-contained. An instance can be moved to a new server in the cluster that shares the storage.

For example, in a four-system cluster, it is possible to have three database instances, or service groups, each at a different version of Oracle. 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 binaries on local storage, would require as many as 12 copies of the binaries (three versions on four systems).

The disadvantage of this approach is that a rolling upgrade of binaries on shared storage is not possible.

\$ORACLE_HOME Directory on the Local Disk

Installing \$ORACLE_HOME on the local disk has the advantage of allowing an upgrade of the Oracle database server running on an offline system while the database server runs on another system. The database server can later be switched to the upgraded system (provided the database is compatible), permitting a minimum amount of downtime.

The disadvantage of this approach is that with a large number of systems, it becomes difficult to maintain the various Oracle installations.

Multiple Oracle Instances (SIDs)

If multiple Oracle instances are defined in a single cluster configuration, the startup profile for each instance must be accessible on all systems listed in the service group's SystemList attribute.

Location of Database Tablespaces

If the tablespaces are created using regular (UFS or VxFS) files, the file systems containing these files must be located on shared disks. Create the same file system mount points on each system.

If you are using raw devices on shared disks for Oracle tablespaces, change the ownership to Oracle dba user and the permissions, or access mode, to 660 on the raw devices storing the Oracle data.

For example, if you are using 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

Configuring the Location of Core Files

The VCS agent framework sets the current directory to `/opt/VRTSvcs/bin/Oracle` before running Oracle agent scripts or programs that execute the Oracle binaries. Since Oracle binaries, which run as the user `oracle`, do not have permission to write to `/opt/VRTSvcs/bin/Oracle`, any “core” files they generate will be lost. We recommend you use the `coreadm (1M)` command available on Solaris to specify the name and location of core files produced by processes that terminate abnormally.

Transparent Listener Failover

To enable Oracle Server clients to reconnect after a system switch without reconfiguring, 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 are using the TCP/IP protocol for Oracle client/server communication, verify that the file `/etc/services` on each system defined in the service group's SystemList attribute contains entries for the service name of the Oracle Net Service.



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 and use it in place of the long filename in the Home attribute in the `main.cf` file. You must create the link on each of the systems in the cluster.

1. Create a soft link to the long pathname. 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 Pfile attributes for both the Oracle resource type and the Netlsnr resource type (refer to the Sample Configuration in “[Sample Configurations](#)” on page 85).

For example:

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

Defining NLS Information

You can define NLS information by defining the appropriate parameters in the Oracle parameter file or by defining the appropriate environment variables in the `EnvFile`. See the description of the `EnvFile` attribute in “[Resource Type Definitions](#)” on page 79.

Defining parameters in the Oracle parameters file affects NLS settings for the Oracle server. Defining environment variables affects the NLS input and output of client utilities.

Hot Backup of Oracle Database and VCS

If a cluster node fails while running a hot backup of an Oracle database, the failover to another node can succeed only if the Oracle resource attribute, AutoEndBkup, is set to a non-zero value. See “[Resource Type Definitions](#)” on page 79 for more information about the Oracle agent attributes.

Otherwise, the database in the backup mode on the failover system cannot be opened and VCS cannot online the Oracle resource group.

The following errors are displayed to indicate this condition:

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

or

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

Before VCS can online the Oracle resource group on the failover system, you must take the datafiles in the database out of the backup mode and shut down the database so that it can be reopened. Refer to the Oracle documentation for instructions on how to change the state of the database files.



Installing Oracle

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

You can install Oracle in the following two ways in a VCS environment:

- ◆ **\$ORACLE_HOME on the shared disk.** Oracle binaries and data, including the datafiles, control files, redo log files, archive log files, and the spfile are installed on shared disks.
- ◆ **\$ORACLE_HOME on the local disk.** Oracle binaries are installed locally on *each* node and the data, including the datafiles, control files, redo log files, archive log files, and the spfile, is installed on shared disks.

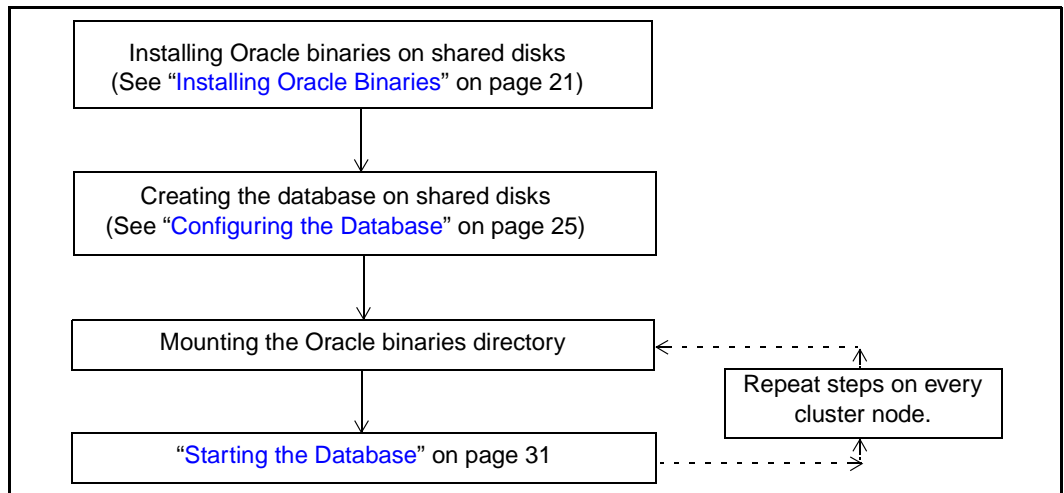
Note When installing Oracle, ensure that the login_id, id_name, group_id, and group_name for the Oracle owner is the same on all the nodes. The user **oracle** and the group **dba** must be local and not Network Information Service (NIS and NIS+) users

Overview of Tasks

This section provides an overview of tasks required to install Oracle in a VCS cluster. These tasks are described in detail in following sections.

\$ORACLE_HOME on Shared Disks

Installing Oracle such that the \$ORACLE_HOME is on shared disks involves the following tasks.

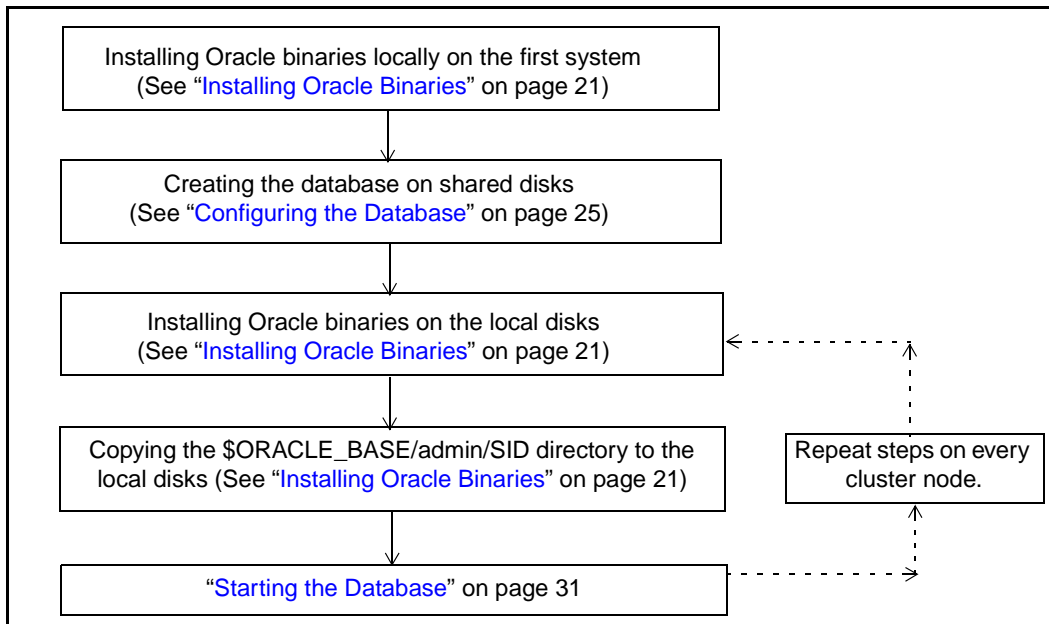


The methodology is to install the Oracle binaries and to create the Oracle database on shared disks during the first installation. Mount the Oracle binaries directory and verify that the database can be started from all nodes.



\$ORACLE_HOME on Local Disks

Installing Oracle such that the \$ORACLE_HOME is on local disks involves the following tasks.



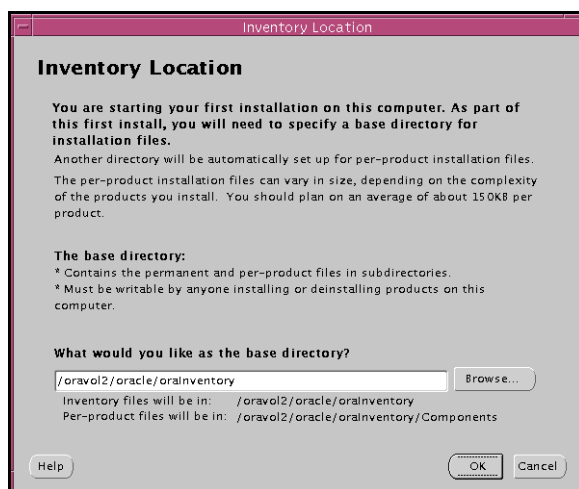
The methodology is to install the Oracle binaries and to create the Oracle database on shared disks during the first installation. This ensures that all Oracle installations are identical and access the database from the same location on the shared disk.

Installing Oracle Binaries

This section describes how to install Oracle 9i for VCS. Note that the screens and options presented by installers for other Oracle versions may differ.

1. Insert the Oracle CD.
2. Set the *DISPLAY* variable, mount the CD drive, and run the Oracle installer as Oracle User.

```
# /mnt/cdrom/runInstaller
```
3. Read the Welcome screen and click **Next**.
4. If this is the first Oracle installation, the **Inventory Location** dialog box is displayed. Specify the location of your base directory and click **OK**.

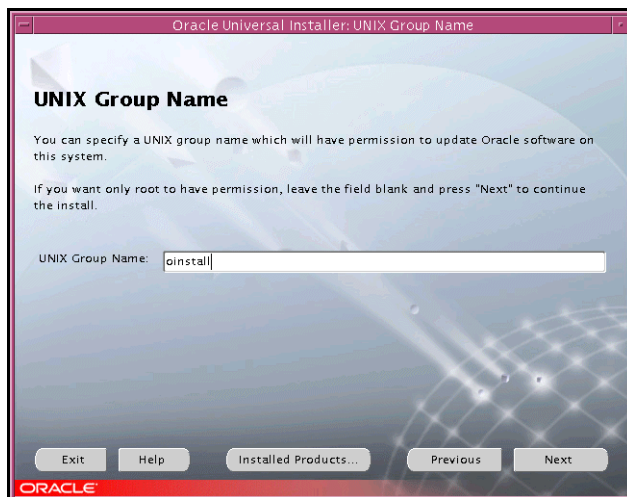


If in your installation, the \$ORACLE_HOME will be on local disks, specify a location on the local disk.

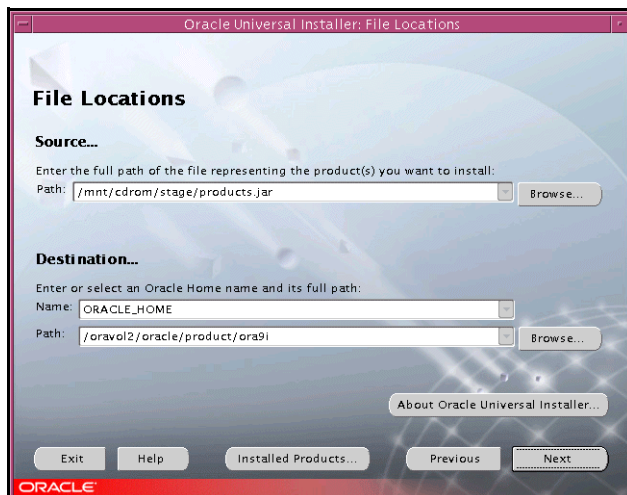
If the \$ORACLE_HOME will be on shared disks, specify a location on the shared disk.



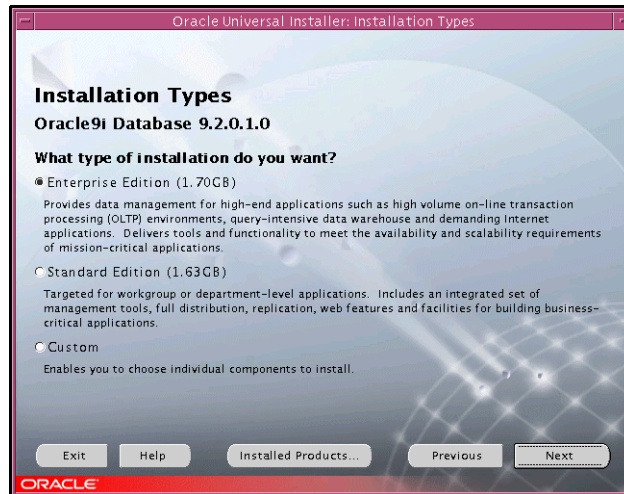
5. On the **UNIX Group Name** dialog box, enter the name of a UNIX group that will have permissions to update the Oracle software and click **Next**.



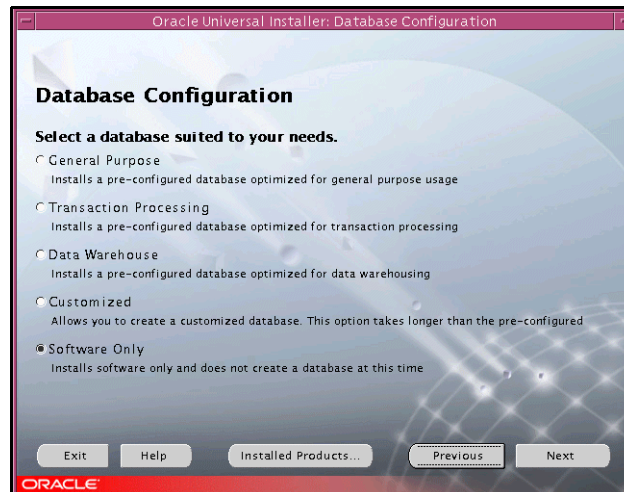
6. The Oracle installer displays a message asking you to run the script `/tmp/orainstRoot.sh`. Run the script and click **Continue** to return to the Oracle installer.
7. On the **File Locations** dialog box, enter or select a name and a path for the Oracle Home and click **Next**.



8. On the **Available Products** dialog box, select the **Oracle 9i Database** option and click **Next**.
9. On the **Installation Types** dialog box, select your installation type and click **Next**.



10. On the **Database Configuration** dialog box, select the **Software Only** option and click **Next**.



11. On the **Summary** dialog box, review your selections. Click **Back** to make changes. Otherwise click **Install**.



12. Click **Next** after the installation is complete.
13. On the **End of Installation** dialog box, click **Exit**.
14. Proceed to the next section to configure the database on shared disks.

Configuring the Database

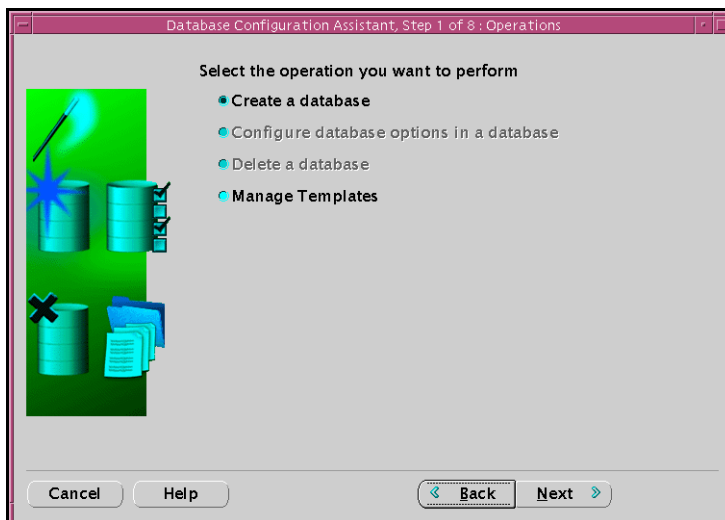
This section instructs you on how to configure an Oracle 9i database on shared disks using the Database Configuration Assistant.

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

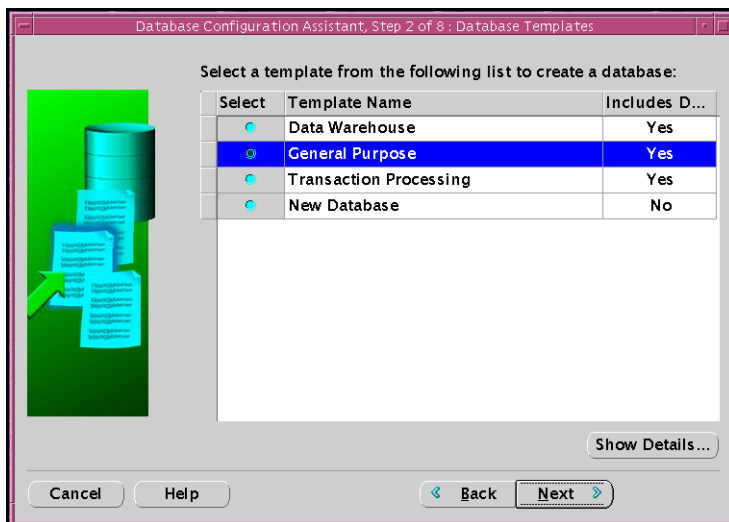
1. Set the *DISPLAY* variable and start the Oracle Database Configuration Assistant as Oracle User.

```
# dbca
```

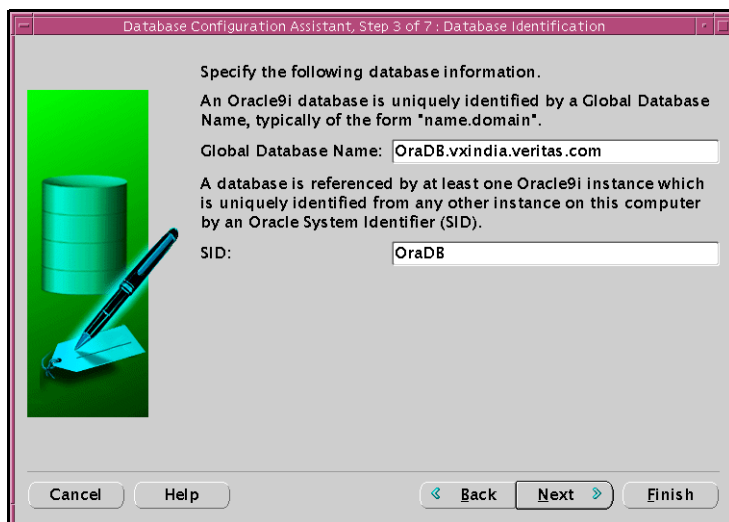
2. Read the Welcome screen, click **Next**.
3. On the **Operations** dialog box, select the **Create a database** option and click **Next**.



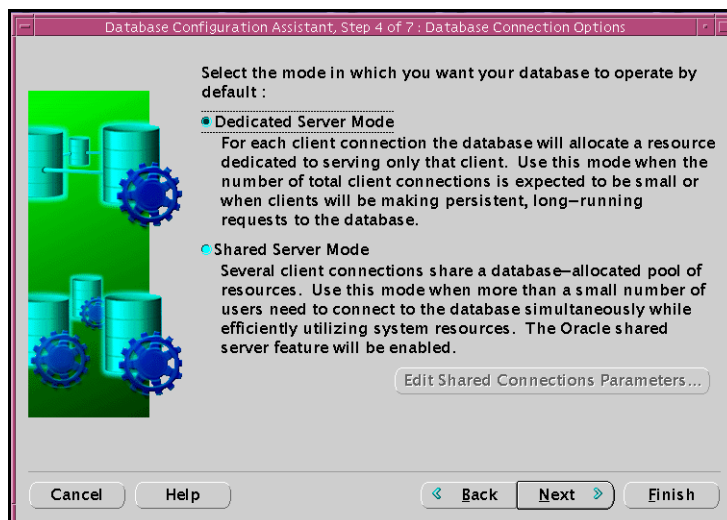
4. On the **Database Templates** dialog box, select a template to create the database and click **Next**.



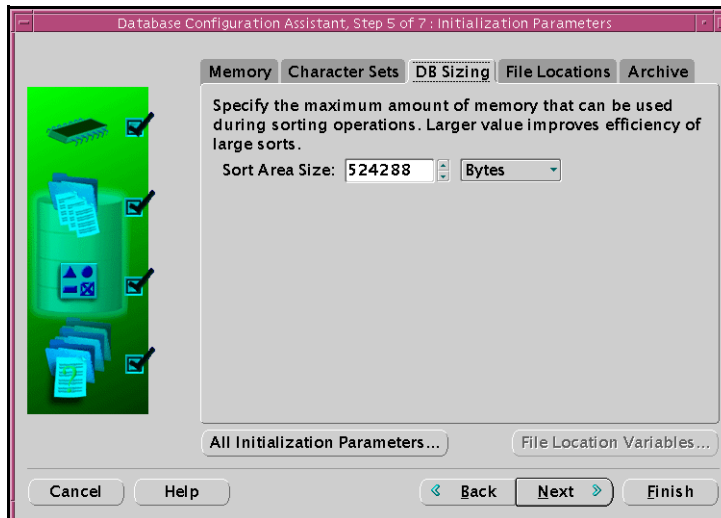
5. On the **Database Identification** dialog box, enter or select a unique name and SID for the global database and click **Next**.



6. On the **Database Connection Options** dialog box, select the **Dedicated Server Mode** or the **Shared Server Mode** and click **Next**.

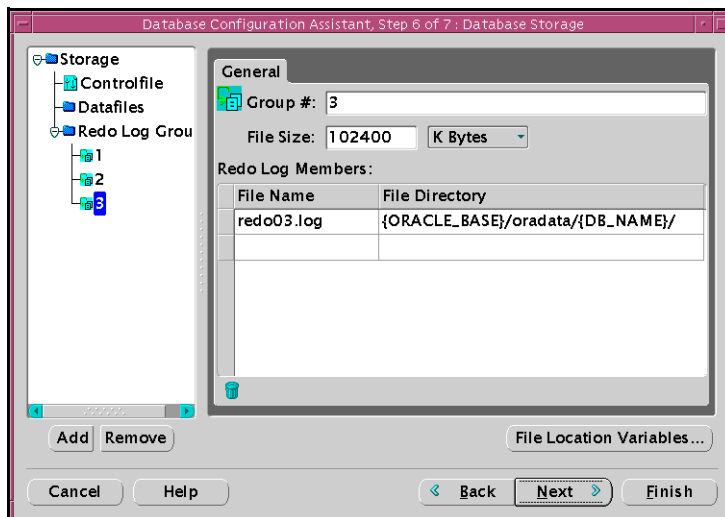


7. On the **Initialization Parameters** dialog box, specify the locations of the archive logs and the initialization parameter files.



- a. If you want to enable archive logs for the database, click the **Archive** tab and select the **Archive Log Mode** check box. In the Archive Log Destination(s) list, enter a path on the shared disk. This ensures that the Archive Log will be created on the shared disk.
- b. Click the **File Locations** tab.
- c. If installing Oracle locally on all systems, make sure that the initialization parameter file and the trace file directories are on the local disk.
- d. Modify other parameters, if desired.
- e. Click **Next**.

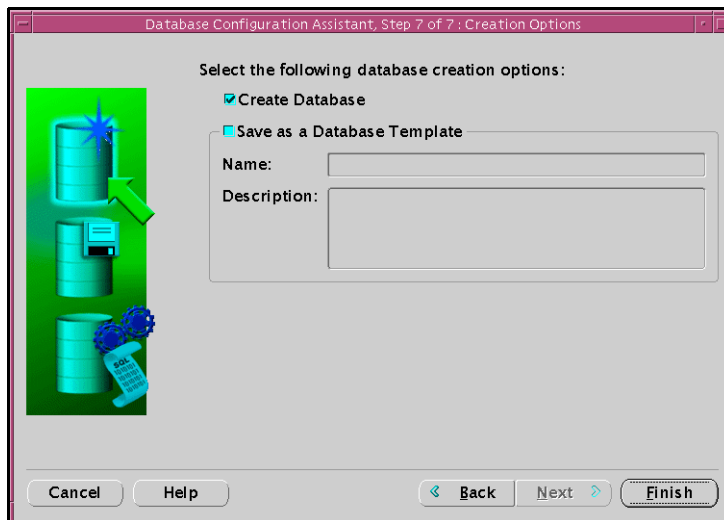
8. On the **Database Storage** dialog box, specify the locations of the Control, Data, and Redo log files.
 - a. In the left pane, expand the Storage folder by clicking the + next to it.
 - b. Click **Controlfile** and click the **General** tab in the right pane.



- c. In the **File Directory** field, enter a path on the shared disk for each control file.
- d. Expand the **Datafiles** folder by clicking the + next to it.
- e. Select the datafile in the left pane and click the **General** tab in the right pane. In the **Name** field, enter a path on the shared disk for each datafile.
- f. Expand the **Redo Log Groups** folder by clicking the + next to it.
- g. Select the Redo Log file in the left pane. In the **File Directory** field in the right pane, enter the path of a directory on the shared disk for each Redo Log file. Specify file sizes for the Redo Log files, if desired.
- h. Click **Next**.



9. On the **Creation Options** dialog box, select the **Create Database** check box and click **Finish**.



10. On the **Summary** dialog box, review your selections and click **OK**.

Note You need to configure and test if the database is accessible on other nodes. If your `$ORACLE_HOME` is on shared disks, mount the Oracle binaries on each node individually, and start the database (see “[Starting the Database](#)” on page 31). If your `$ORACLE_HOME` is on local disks, install Oracle locally on other systems (see “[Installing Oracle Binaries](#)” on page 21) and start the database on each system, individually (see “[Starting the Database](#)” on page 31).

Copying the \$ORACLE_BASE/admin/SID Directory

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

Copy the directory \$ORACLE_BASE/admin/SID from the shared disk to the local disk. The variable SID represents the database instance.

Refer to Oracle's OFA Structure for more information.

Starting the Database

Mount the database directory and start the database on each node. You can verify access to the database by running an SQL query.





Installing the Oracle Agent

3

This chapter describes how to install the Oracle agent in a VCS cluster. If you have an earlier version of the agent installed, refer to “[Upgrading the Agent](#)” on page 35.

Installing the Agent Software

The VCS enterprise agent for Oracle is installed from the CD. You must install the Oracle Enterprise Agent on all systems that will host the Oracle service group.

Prerequisites

- ✓ Make sure that VCS is installed in the cluster.
- ✓ If a previous version of the agent is installed, uninstall it.
- ✓ Install and configure the Oracle Server for Solaris and the add-on Listener. For more information, see the Oracle documentation.

Installation Instructions

1. Log in as `root`.
2. Insert the CD into a drive connected to your system.

If you are running Solaris volume-management software, the software automatically mounts the CD as `/cdrom/cdrom0`. Type the following command to install the agent:

```
# cd /cdrom/cdrom0
# pkgadd -d .
```



If you are not running Solaris volume-management software, you must mount the CD 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.

```
# cd /cdrom
# pkgadd -d .
```

This installs the following packages:

- VRTSvcsor—the agent binaries
- VRTScsocw—the agent configuration wizard

Note Repeat steps 1 and 2 on each system that will become part of the Oracle service group.

Upgrading the Agent

This section describes how to upgrade from the VCS enterprise agent 3.5 for Oracle in a VCS cluster.

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

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

2. Follow step a through step c on all systems on which the Oracle agent was installed.

- a. Remove the Oracle agent.

```
# pkgrm VRTSvcsor
```

- b. Delete the file `/etc/VRTSvcs/conf/config/OracleTypes.cf`.

- c. Install the VCS enterprise agent 4.0 for Oracle. See “[Installing the Agent Software](#)” on page 33 for instructions.

3. Copy the file `OracleTypes.cf` from the directory `/etc/VRTSvcs/conf/` to the `/etc/VRTSvcs/conf/config` directory.

4. 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 [Appendix A](#) for information about the attributes in this version of the Oracle agent.

5. Verify the configuration

```
# hacf -verify config
```

6. Start VCS on local node first.

7. Start VCS on other nodes.





Configuring the Agent

4

Configuring the Oracle enterprise agent involves creating the Oracle service group, its resources, and defining attribute values for the configured resources. VCS provides several ways to configure an enterprise agent, including the agent configuration wizard, Cluster Manager (Java Console), and the command line.

You must have administrator privileges to create and configure a service group.

Before configuring the agent, review the supported Oracle configurations, described under “[Oracle Configurations in VCS](#)” on page 38. Review the Oracle resource type and the attribute definitions of the Oracle agents, described in “[Resource Type Definitions](#)” on page 79. For sample configuration files of the Oracle service group, see “[Sample Configurations](#)” on page 85.

Prerequisites

- ✓ Install and configure Oracle identically on all cluster systems. (For more information, see “[Installing and Configuring Oracle](#)” on page 13.)
- ✓ Verify that the Oracle agent is installed on all cluster systems. (For more information, see “[Installing the Oracle Agent](#)” on page 33.)



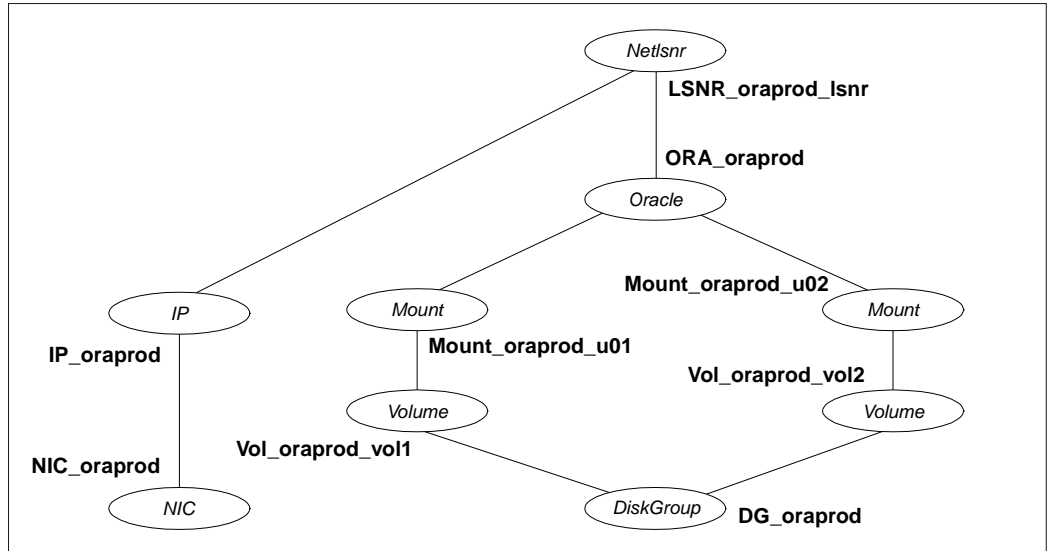
Oracle Configurations in VCS

You can set up Oracle in different ways in a VCS environment.

- ◆ **Single Oracle instance configuration:** Provides high availability for an Oracle instance.
- ◆ **Multiple Oracle instance configuration:** Provides high availability for several Oracle instances. The Oracle instances can be configured to share a common listener, or can have a dedicated listener per instance.
- ◆ **Configuration with Shared Server support:** Provides high availability for an Oracle instance having shared server support.

Single Oracle Instance Configuration

The following example describes a typical service group configured to monitor the state of an Oracle instance in a VCS cluster.



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.

If your configuration does not use VERITAS Volume Manager, use the `DiskReservation` resource type to configure shared storage instead of the `DiskGroup` and `Volume` resource types.



▼ To configure an Oracle instance in VCS

1. Review the Oracle and Netlsnr resource types and their attribute definitions, described in [“Resource Type Definitions”](#) on page 79.
2. Configure Oracle for VCS. This involves configuring the Oracle files listener.ora and tnsnames.ora as per VCS requirements. This section describes the changes required in these files, depending on your Oracle configuration.

- a. For clients to connect to the fail over 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.

In the following example, assuming 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)
    )
  )
```

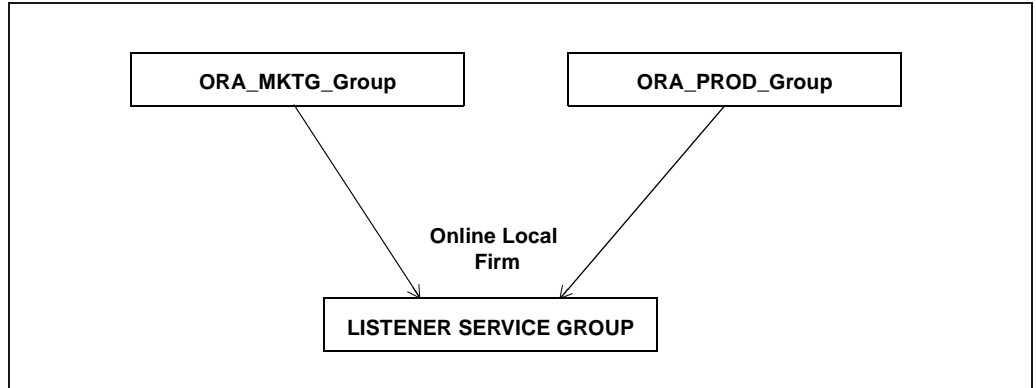
- b. Perform additional configuration tasks specific to the listener configuration. In the file listener.ora located at \$TNS_ADMIN, edit the "Host=" line in the ADDRESS_LIST section and add the name of the high availability address for the service group, in this case, oraprod.

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

3. Create the Oracle service groups using the Agent Configuration wizard, Cluster Manager (Java Console,) or the command-line. For information on using the wizard, see [“Configuring the Agent Using the Agent Configuration Wizard”](#) on page 49. For information on using Cluster Manager (Java Console), see [“Configuring the Agent Using Cluster Manager \(Java Console\)”](#).
4. Bring the Oracle service group online. See [“Bringing the Service Group Online”](#) on page 69 for instructions.

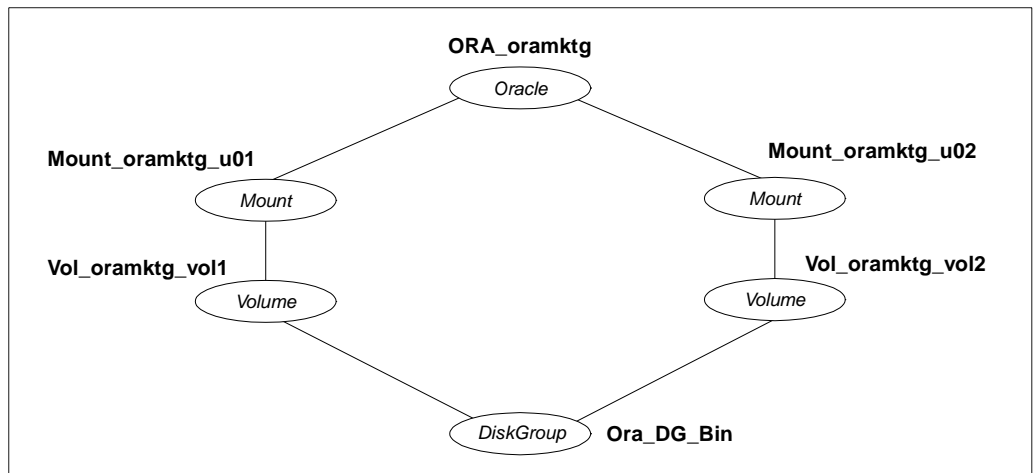
Multiple Oracle Instances (Single Listener)

This configuration has a service group for each Oracle instance. The Listener too is configured in a separate service group.

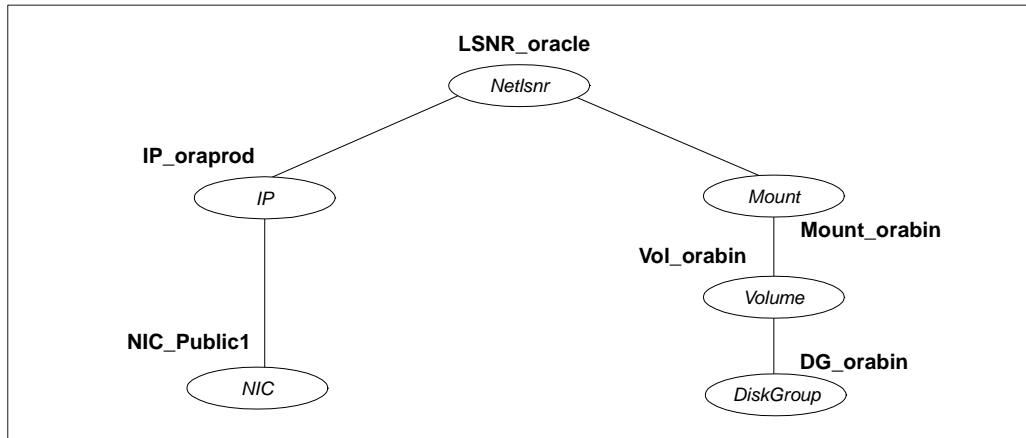


The Oracle service groups are made dependent on the Listener service group using an Online Local Firm dependency.

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.



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.



The Oracle server can be started after the Listener service group and the resources in the Oracle service group are brought online.

If your configuration does not use VERITAS Volume Manager, use the DiskReservation resource type to configure shared storage instead of the DiskGroup and Volume resource types.

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.

▼ To configure multiple Oracle instances (single listener)

1. Review the Oracle and Netlsnr resource types and their attribute definitions, described in [“Resource Type Definitions”](#) on page 79.
2. Configure Oracle for VCS. This involves configuring the Oracle files `listener.ora` and `tnsnames.ora` as per VCS requirements. This section describes the changes required in these files, depending on your Oracle configuration.

- a. For clients to connect to the fail over 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.

In the following example, assuming 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)
    )
  )
```

- b. Perform additional configuration tasks specific to the listener 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 = 1522))
      )
    )
  )
```

3. Create the Oracle and Listener service groups using Cluster Manager (Java Console.) See [“Configuring the Agent Using Cluster Manager \(Java Console\)”](#) on page 59 for instructions. You can also use the command-line to create the service group.
4. Bring the Oracle service group online. See [“Bringing the Service Group Online”](#) on page 69 for instructions.



Multiple Oracle Instances (Multiple Listeners)

This configuration has several single-instance configurations. Each Oracle instance is configured in a separate service group. See the section describing the configuration for “[Single Oracle Instance Configuration](#)” on page 39 for more information.

▼ To configure multiple Oracle instances (multiple listeners)

1. Review the Oracle and Netlsnr resource types and their attribute definitions, described in “[Resource Type Definitions](#)” on page 79.
2. Configure Oracle for VCS. This involves configuring the Oracle files listener.ora and tnsnames.ora as per VCS requirements. This section describes the changes required in these files, depending on your Oracle configuration.
 - a. For clients to connect to the fail over 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)
    )
  )
```

- b.** In the file `listener.ora`, create independent configurations for each listener.

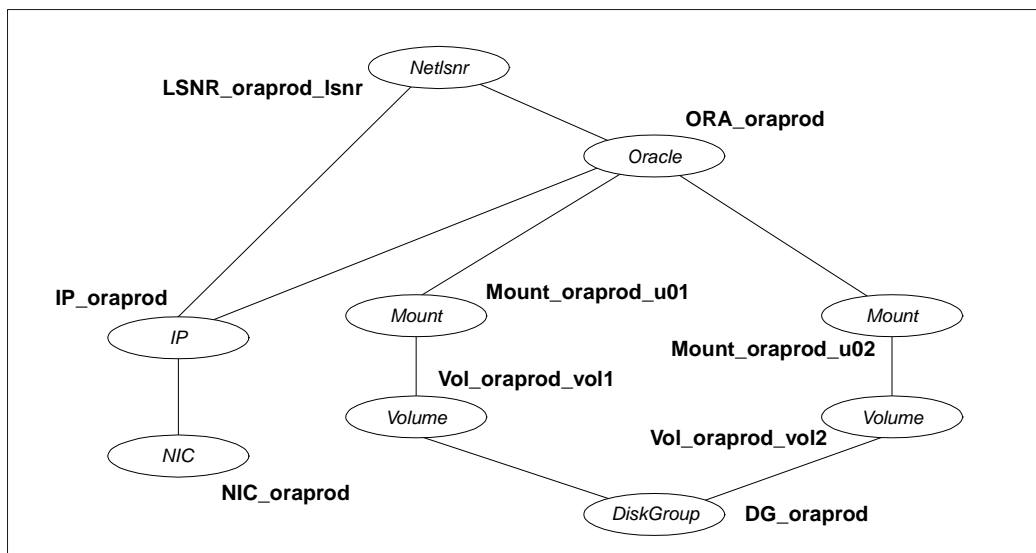
```
LISTENER_PROD =  
  (DESCRIPTION_LIST =  
    (DESCRIPTION =  
      (ADDRESS_LIST =  
        (ADDRESS = (PROTOCOL = TCP)(HOST = oraprod)(PORT = 1521))  
      )  
    )  
  )  
  
LISTENER_MKTG =  
  (DESCRIPTION_LIST =  
    (DESCRIPTION =  
      (ADDRESS_LIST =  
        (ADDRESS = (PROTOCOL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

- 3.** Create the Oracle service groups using the Agent Configuration wizard, Cluster Manager (Java Console,) or the command-line. For information on using the wizard, see [“Configuring the Agent Using the Agent Configuration Wizard”](#) on page 49. For information on using Cluster Manager (Java Console), see [“Configuring the Agent Using Cluster Manager \(Java Console\)”](#) on page 59.
- 4.** Bring the Oracle service group online. See [“Bringing the Service Group Online”](#) on page 69 for instructions.



Oracle With Shared Server Support

The following example describes a typical service group configured to monitor Oracle 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.

If your configuration does not use VERITAS Volume Manager, use the `DiskReservation` resource type to configure shared storage instead of the `DiskGroup` and `Volume` resource types.

▼ To configure Oracle with shared server support

1. Review the Oracle and Netlsnr resource types and their attribute definitions, described in [“Resource Type Definitions”](#) on page 79.
2. Configure Oracle for VCS. This involves configuring the Oracle files `listener.ora` and `tnsnames.ora` as per VCS requirements. This section describes the changes required in these files, depending on your Oracle configuration.

- a. For clients to connect to the fail over 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.

In the following example, assuming 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)
    )
  )
```

- b. Perform additional configuration tasks specific to the listener 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))
      )
    )
  )
```

3. In the initialization parameter file, for the `mts_dispatchers` (called dispatchers in Oracle 9i versions and above) parameter, set the host name for all TCP protocol address dispatchers to the virtual IP address for the service group.

In the following example from an Oracle 8i configuration, the host name for the dispatcher is set to `oraprod`.

```
mts_dispatchers =
  "(ADDRESS = (PROTOCOL = TCP)(HOST = oraprod))"
```



```
(MUL=ON)(TICK=15)(POOL=true)"
```

In the following example from an Oracle 9i configuration, the host name for the dispatcher is set to oraprod.

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

4. 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`.

5. 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))
```

6. Create the Oracle service groups using the Agent Configuration wizard, Cluster Manager (Java Console,) or the command-line. For information on using Cluster Manager (Java Console), see [“Configuring the Agent Using Cluster Manager \(Java Console\)”](#) on page 59.
7. Bring the Oracle service group online. See [“Bringing the Service Group Online”](#) on page 69 for instructions.

Configuring the Agent Using the Agent Configuration Wizard

VCS provides an Agent Configuration wizard that guides you through the process of configuring the agent. The wizard creates and modifies Oracle service groups.

This section describes the steps required to create a new Oracle service group using the wizard. To modify an existing service group, see “[Modifying the Agent Configuration](#)” on page 72.

Before configuring the Oracle agent, review the resource type and the attribute definitions of the Oracle and Netlsnr agents, described in “[Resource Type Definitions](#)” on page 79. For sample configuration files and resource dependency graphs of the Oracle service group, see “[Sample Configurations](#)” on page 85.

Prerequisites

- ✓ Make sure that the Oracle instances and listeners to be configured are running. All listeners to be configured must listen to the same virtual IP address.

Note For the default listener, the configuration wizard requires the listener parameter file, `listener.ora`, to reside in `$ORACLE_HOME/network/admin`. No such condition applies for non default listeners.

- ✓ The Oracle files (control, data, redo-log and archive files) must be on shared disks.
- ✓ The IP addresses and host names specified in the files `listener.ora` and `tnsnames.ora` must be the same.
- ✓ If detail monitoring is to be used for a database instance, the table used for detail monitoring must be set up, with user and password assigned.

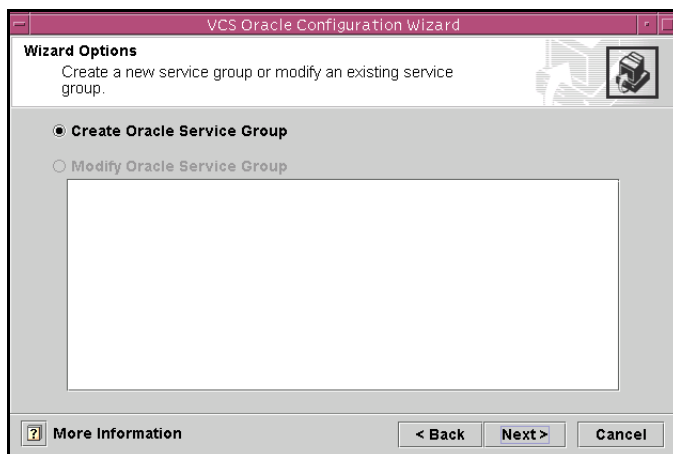
Instructions

1. Set the `DISPLAY` variable, start the Oracle configuration wizard as `root`.

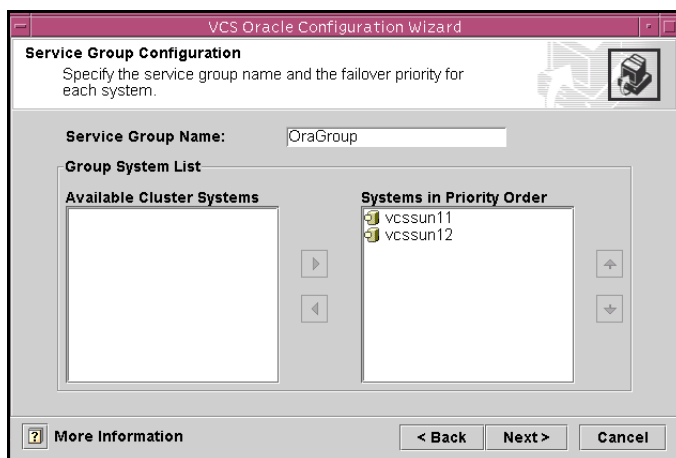
```
# hawizard oracle
```
2. Read the information on the Welcome screen and click **Next**.



3. On the **Wizard Options** dialog box, select the **Create Oracle Service Group** option and click **Next**.



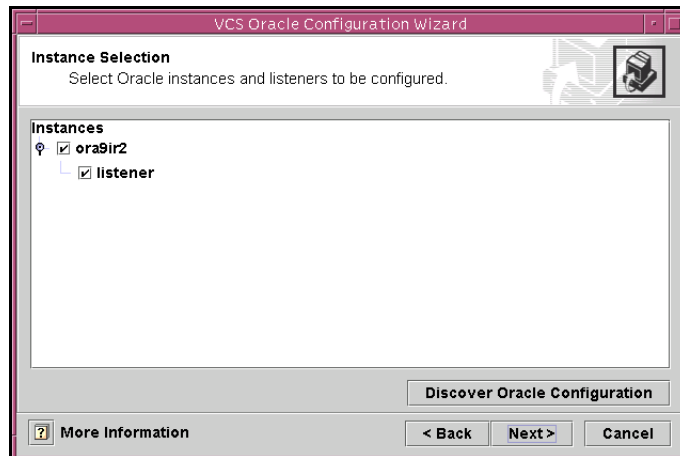
4. On the **Service Group Configuration** dialog box, specify information about the service group.



- a. Enter a name for the Oracle service group.
- b. In the **Available Cluster Systems** box, select the systems on which to configure the service group and click the right-arrow icon to move the systems to the service group's system list.

To remove a system from the service group's system list, select the system in the **Systems in Priority Order** box and click the button with the left arrow icon.

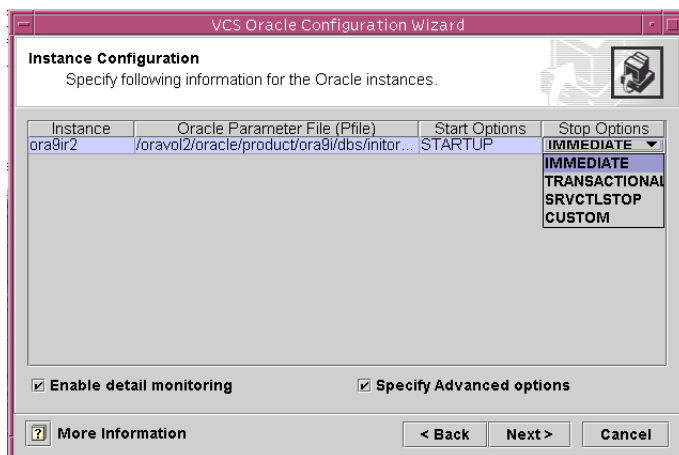
- c. To change a system's priority in the service group's system list, select the system in the **Systems in Priority Order** box and click the buttons with the up and down arrow icons. The system at the top of the list has the highest priority while the system at the bottom of the list has the lowest priority.
 - d. Click **Next**.
5. On the **Instance Selection** dialog box, select the Oracle instances to be configured along with their listeners. Click **Discover Oracle Configuration**, if required.



Click **Next**.



6. On the **Instance Configuration** dialog box, specify information for all selected instances.



- a. Verify the location of the Oracle Parameter File.
- b. Choose Start and Stop options, if desired.
- c. Select the **Enable detail monitoring** and **Specify Advanced options** check boxes, if desired.
- d. Click **Next**.

7. If you chose to monitor the database in detail, the **Detail Monitoring** dialog box is displayed.

Select	Sid	User	Password	Table
<input checked="" type="checkbox"/>	ora9ir2	thor	*****	table1

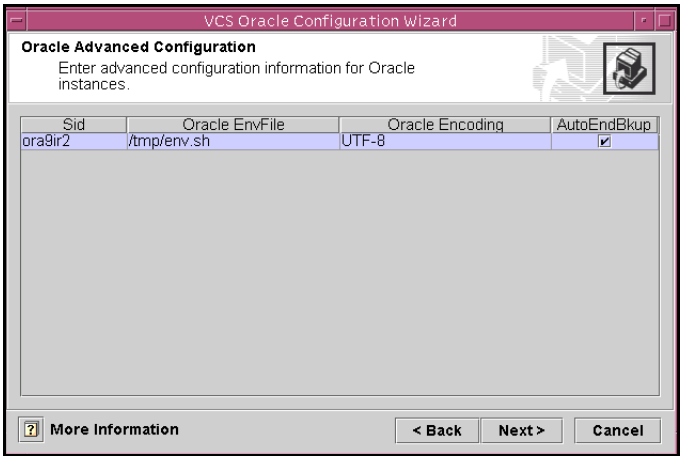
- a. In the **Select** column, select the check box corresponding to the database to be monitored in detail.
- b. Enter a valid user name for the database.
- c. Enter a valid password for the database user.

Note Do not encrypt passwords when entering them through the Agent Configuration Wizard; the wizard takes care of encrypting passwords.

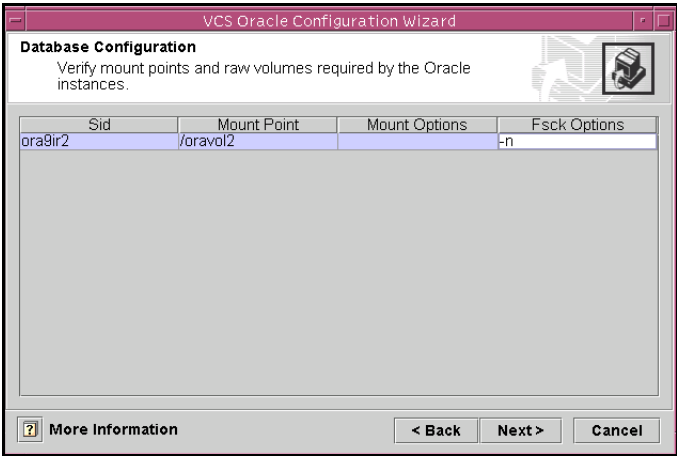
- d. In the **Table** column, enter the name of a table that will be queried to validate the status of the database.
- e. Click **Next**.



8. If you chose to specify advanced options, the **Oracle Advanced Configuration** dialog box is displayed.



- a. Enter the location of the Oracle Envfile and the encoding and select the AutoEndBkup check box, if desired.
- b. Click **Next**.
9. On the **Database Configuration** dialog box, verify the mount points required by each database, specify the Mount and Fsck options, and click **Next**.



10. On the **Listener Configuration** dialog box, configure the listeners for the databases.

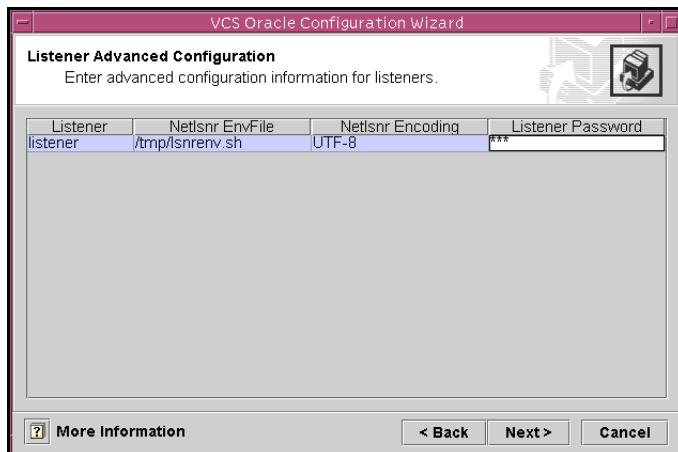
The screenshot shows the 'Listener Configuration' dialog box from the 'VCS Oracle Configuration Wizard'. The title bar reads 'VCS Oracle Configuration Wizard'. The main title is 'Listener Configuration' with the subtitle 'Configure the network details for the listeners.' Below this, there are two text input fields: 'Address' with the value '10 . 212 . 102 . 218' and 'Netmask' with the value '255 . 255 . 254 . 0'. A table with two columns, 'System' and 'Device', contains two rows: 'vcssun11' and 'vcssun12', both with 'hme0' selected in the 'Device' column. At the bottom, there are two checked checkboxes: 'Enable detail monitoring' and 'Specify Advanced options'. The bottom bar includes a 'More Information' link with a question mark icon, and three buttons: '< Back', 'Next >', and 'Cancel'.

System	Device
vcssun11	hme0
vcssun12	hme0

- a. Verify the virtual IP address and Netmask.
- b. For each system, select a device.
- c. Select the **Enable detail monitoring** and **Specify Advanced options**, if desired.
- d. Click **Next**.



11. If you chose to specify advanced options for the listeners, the **Listener Advanced Configuration** dialog box is displayed.

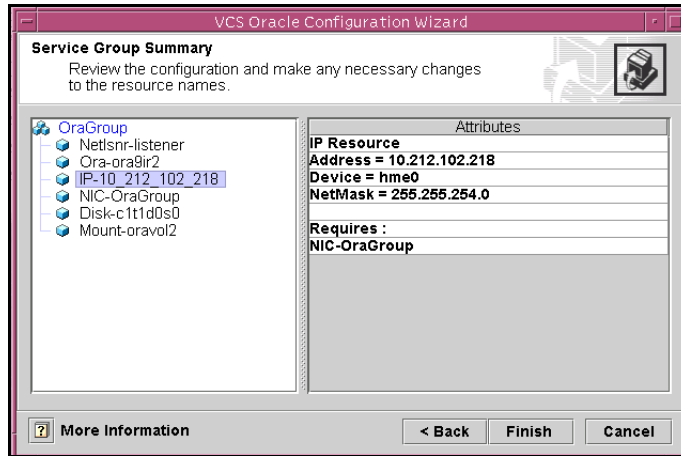


For each listener, enter the path of the Envfile and encoding, if desired. Enter a valid password for the listener.

Click **Next**.

Note Do not encrypt passwords when entering them through the Agent Configuration Wizard; the wizard takes care of encrypting passwords.

12. On the Service Group Summary dialog, review your configuration.



Click on a resource to view its attributes and their configured values in the **Attributes** box.

- a. Change names of resources, if desired; the wizard assigns unique names to resources based on their respective name rules.

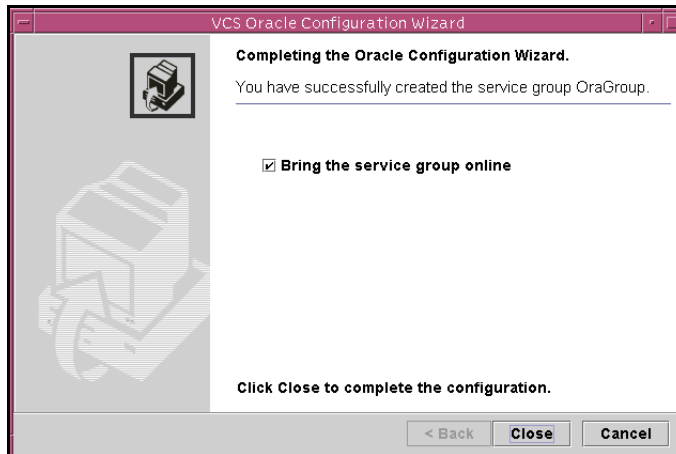
To edit a resource name, select the resource name and click on it, press Enter after editing each attribute.

- b. Review your configuration and Click **Finish**.

The wizard starts running commands to create the Oracle service group. Various messages indicate the status of these commands.



13. On the **Completing the Oracle Configuration wizard** dialog box, select the **Bring the service group online** check box to bring the service group online on the local system.



Click **Close**.

The Oracle service group is created in your cluster.

Configuring the Agent 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.

Importing the OracleTypes.cf File

To use the Oracle template, import the `OracleTypes.cf` file to the VCS engine by using 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 file
`/etc/VRTSvcs/conf/OracleTypes.cf`
4. Click **Import**.
5. Save the configuration.

At this point, the Oracle types have been imported to the VCS engine. The Oracle agent can be configured without interrupting or stopping VCS.



Creating a New Service Group

1. Make sure the Oracle type definition file `OracleTypes.cf` is imported in your configuration.
2. Launch the Cluster Configuration wizard using any of the following two ways:
 - 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.
 - a. Enter the name of the service group.
 - b. On the **Available Systems** box, select the systems on which to configure the service group.
 - c. 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.
 - d. 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.
 - e. Click **Next**.
5. On the **Would you like to use a template to configure the service group?** dialog box, click **Next**.
6. On 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.
9. A window opens listing the service group's resources and their associated attributes. 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.

Editing Resource attributes

Configure the Oracle service group by editing attributes of resources in the service group.

1. Select the resource from the list on the left. Its attributes appear in the right pane.
2. Select the attribute to be modified.
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. Click **Finish** to quit the wizard.



Configuring the Agent Using the Command-Line

The VCS enterprise 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.

Follow the instructions below to configure the agent using the sample configuration file:

1. Log in to sysa as `root`.
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. Copy the Oracle types configuration file into place:

```
# cp /etc/VRTSvcs/conf/OracleTypes.cf
   /etc/VRTSvcs/conf/config/OracleTypes.cf
```

6. Edit the `main.cf` file. The file `/etc/VRTSvcs/conf/sample_oracle/main.cf` can be used for reference:

- a. Add an “include” line for the `OracleTypes.cf` file:

```
# include "OracleTypes.cf"
```

- b. Create the Oracle and Netlsnr resources. You can refer to the examples shown in the file `/etc/VRTSvcs/conf/sample_oracle/main.cf`.

- c. Edit the default attributes to match the parameters in your configuration. See “[Resource Type Definitions](#)” on page 79 for more information about the agent attributes.

- d. Assign dependencies to the newly created resources. Refer to the sample file `/etc/VRTSvcs/conf/sample_oracle/main.cf`, and the sample dependency graphs in “[Sample Configurations](#)” on page 85. (See the *VERITAS Cluster Server User’s Guide* for more information on assigning dependencies.)

7. Save and close the file.
8. Verify the syntax of the file `/etc/VRTSvcs/conf/config/main.cf`:

```
# cd /etc/VRTSvcs/conf/config/  
# hacf -verify .
```
9. Start the VCS engine:

```
# hastart
```
10. Verify that all Oracle service group resources are brought online:

```
# hagr -state
```
11. Take the service group offline and verify that all resources are stopped:

```
# hagr -offline service_group -sys system_name  
# hagr -state
```
12. Bring the service group online again and verify that all resources are available:

```
# hagr -online service_group -sys system_name  
# hagr -state
```
13. Start the VCS engine on sysb:

```
# hastart
```
14. Switch the Oracle service group to sysb:

```
# hagr -switch service_group -to sysb
```
15. Verify that all Oracle service group resources are brought online on sysb:

```
# hagr -state
```
16. On all the systems, 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
```



Encrypting Passwords

VCS provides a utility to encrypt database user passwords and listener passwords.

Encrypt the `Pword` attribute in the Oracle agent and the `LsnrPwd` attribute in the `Netlsnr` agent before configuring these attributes.

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 `vcseencrypt` utility before configuring the `LsnrPwd` attribute. This password, when decrypted should be of the same format as stored in the `listener.ora` file.

Note You need not encrypt passwords when using the configuration wizard or the VCS Cluster Manager (Java Console) to configure attributes.

1. From the path `$VCS_HOME/bin/`, run the `vcseencrypt` utility.

a. Type the following command:

```
# root@vcssun28:/opt/VRTSvcs/bin> vcseencrypt -agent
```

b. The utility prompts you to enter the password twice. Enter the password and press Return.

```
# Enter New Password:
```

```
# Enter Again:
```

2. The utility encrypts the password and displays the encrypted password.

3. Enter this encrypted password as the value for the attribute.

Copy the encrypted password for future reference.

Additional Considerations with Basic Monitoring

Basic monitoring of Oracle processes is user-specific. As a result, an Oracle instance started under the context of another user cannot be detected as online. For example, if an Oracle instance is started under the user “oraVRT” and the agent is configured for a user “oracle”, the agent will not detect the instance started by “oraVRT” as online.

This could lead to situations where:

- ◆ issuing a command to online a resource on a system might online an already running instance on that system (or any other system).
- ◆ within a failover service group, when the administrator onlines an Oracle resource on a system and if the Oracle instance is online on any other system within a cluster, the instance would come up. However, the database does not get mounted. In such circumstances, this failure is detected only by in-depth monitoring. If in-depth monitoring is not configured, then such a conflict would go undetected.

Hence, VERITAS recommends that instances started outside VCS control be configured with the correct Owner attribute corresponding to the OS user for that instance.

Setting Up Detail Monitoring

The VCS enterprise agent for Oracle provides two levels of application monitoring: primary (basic monitoring) and secondary (detail monitoring). In the basic monitoring mode, the agent monitors the Oracle processes to verify that they are continuously active.

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

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 VCS enterprise agent for Oracle. The scripts are available under the directory `/opt/VRTSvcs/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 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 following example, 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.

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 `svrmgrl` or `sqlplus` utility to set up a database table:

```
$ORACLE_HOME/bin/svrmgrl

or

$ORACLE_HOME/bin/sqlplus /nolog
```

6. As the database administrator, issue the following statements at the `svrmgr1` or `sqlplus` prompt to create the test table:

```
connect / as sysdba

create user <User>
identified by <Pword>
default tablespace USERS
temporary tablespace TEMP
quota 100K on USERS;

grant create session to <User>;

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

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

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

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

8. Detail monitoring for the Oracle resource can be enabled by 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 -modify OracleResource DetailMonitor 1
haconf -dump -makero
hagrp -unfreeze service_group
```

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



Enabling and Disabling Detail Monitoring for Oracle

To enable detail monitoring, set the DetailMonitor attribute to 1.

```
hares -modify OracleResource DetailMonitor 1
```

To disable detail monitoring, set the DetailMonitor attribute to 0.

```
hares -modify OracleResource DetailMonitor 0
```

Setting Up Detail Monitoring for Netlsnr

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

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"  
haconf -dump -makero  
hagrp -unfreeze service_group
```

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

```
haconf -makerw  
hagrp -freeze service_group  
hares -modify SqlResource MonScript ""  
haconf -dump -makero  
hagrp -unfreeze
```

Administering the Oracle Service Group

5

This chapter describes some administrative tasks you can perform on the Oracle service group, including bringing the service group online, taking it offline, switching it, and modifying the service group configuration. The chapter also describes how to disable the agent and to remove it.

Bringing the 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**. This enables 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*).



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.

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*).

Taking the 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*).

Disabling the Agent

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

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:

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

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

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

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

3. Stop the agent on the system by entering:

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

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.

You can now remove the service group, the resource type, or both from the VCS configuration. See the chapter on reconfiguring VCS from the command line in the *VERITAS Cluster Server User's Guide* for more information.



Modifying the Agent Configuration

You can dynamically configure the VCS enterprise agent for Oracle in several ways, including the configuration wizard, the command-line interface, Cluster Manager Java Console, and the Web Console.

This section describes the process using the cluster configuration wizard.

1. Start the Oracle Configuration wizard.

```
# hawizard oracle
```

2. Read the information on the Welcome screen and click **Next**.

3. On the **Wizard Options** dialog box, select the **Modify service group** option, select the service group to be modified, and click **Next**.

4. Follow the wizard instructions and make modifications as per your configuration. See [“Configuring the Agent Using the Agent Configuration Wizard”](#) on page 49 for more information about the configuration wizard.

Removing the Agent

Removing the agent involves removing the agent configuration wizard and the agent files *from each system*.

1. Remove the agent configuration wizard.

```
# pkgrm VRTScsow
```

2. Remove the agent. Answer prompts accordingly.

```
# pkgrm VRTSvcSor
```


Troubleshooting

This chapter lists error messages associated with the VCS enterprise agent. Messages include the description and a recommended solution, if available.

Error Messages Common to the Oracle and Netlsnr Agents

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 to the correct full path name of the Oracle home.
Oracle home directory %s does not exist	The string specified for the Home attribute in the Oracle or Netlsnr type is incorrect. Solution: Set the Home attribute to the correct full path name of the Oracle home.
File %s is not a valid text file	The file specified by the EnvFile attribute for sourcing the environment variables is not present, not readable, or is not a text file. Solution: Set the EnvFile attribute 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.



Message	Description and Solution
Process %s restarted	Warning message to indicate that the PID for the Oracle process specified is different than the one registered by the previous monitor cycle.
Execution of script returned %s	The detailed monitoring script returns the output specified in the message when the status is offline or unknown. Solution: Take corrective action for the error that has occurred.
Monitor procedure %s returned %s	MonScript failed to execute correctly. Solution: Debug MonScript to assess the exact cause of failure.
Monitor procedure %s did not exit, return value is %s	Internal error while executing MonScript. Solution: Contact Technical Support for further assistance.
Source file for environment variables %s is not a valid text file	The file specified by the EnvFile attribute for sourcing the environment variables is not present, not readable, or is not a text file. Occurs in MonScript file. Solution: Set the EnvFile attribute to the correct full path name. Ensure that the file format is valid.
No owner for Oracle executables was specified	The Owner attribute in the Oracle type has not been set. Solution: Set the Owner attribute to the correct owner of the database binaries.
Invalid owner %s for Oracle executables was specified	The OS user specified by the Owner attribute is invalid. Solution: Set the Owner attribute to the correct owner of the database binaries.

Error Messages Specific to the Oracle Agent

Message	Description and Solution
No SID specified	The Sid attribute in the Oracle type has not been set. Solution: Set the Sid attribute to the correct database instance.
No initialization parameter file specified for startup	The Pfile attribute in the Oracle type has not been set. Solution: Set the Pfile attribute to the correct full path name corresponding to the instance specified by Sid attribute.
svrmgrl/sqlplus not found in %s/bin	The client utilities svrmgrl or sqlplus are not found in the \$ORACLE_HOME/bin directory. Solution: Verify that the Oracle home has been correctly specified and that these executables are present.
Oracle database is in backup mode	Warning message to indicate that the database is in backup mode. Solution: If the AutoEndBkup attribute is set then the database files that are being backed up will be taken out of backup mode and the database will be opened.
Oracle %s failed to stop	Warning message to indicate that Shutdown immediate or Shutdown abort commands were not successful in closing the Oracle instance, in the clean or offline entry point.
Oracle instance %s not running	Warning message to indicate that the database instance was not running even before the clean or offline entry points were executed. Solution: No action required.
Oracle (%s) kill TERM %s	Warning message to indicate that the Oracle processes would be signaled with SIGTERM. Solution: No action required.



Message	Description and Solution
Oracle (%s) kill KILL %s	Warning message to indicate that the Oracle processes would be signaled with SIGKILL. Solution: No action required.
Oracle (%s) failed to stop even after SIGKILL	Oracle processes could not be terminated even with SIGKILL. Solution: Contact your Oracle Administrator for assistance.
Oracle (%s) attempting cleanup of Oracle IPC resources	Warning message to indicate that the IPC resources are being cleaned.
Database in QUIESCING/QUIESCED mode	Warning message to indicate that database is in QUIESCING or QUIESCED mode.
Database in RESTRICTED mode	Warning message to indicate that database is in RESTRICTED mode.
Database in SUSPENDED state	Warning message to indicate that database is in SUSPENDED state.
Resource %s - monitor procedure did not complete within the expected time.	Refer to Oracle's alert log for more information. One 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 in the file init.ora to TRUE.
Custom script \$VCS_HOME/bin/Oracle/start_cust om_%s.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 and the specified location and has valid permissions.
Custom script \$VCS_HOME/bin/Oracle/shut_cust om_%s.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.

Error Messages Specific to the Netlsnr Agent

Message	Description and Solution
Cannot open process directory.	The agent could not process the <code>/proc</code> entries in the particular monitor cycle. Solution: No action required.
Listener process %s not running	Warning message to indicate that the Listener process was not running even before the clean or offline entry points were executed. Solution: No action required.
Listener %s kill TERM %s	Warning message to indicate that the Listener process would be signaled with SIGTERM. Solution: No action required.
Listener %s kill KILL %s	Warning message to indicate that the Listener process would be signaled with SIGKILL. Solution: No action required.
Listener %s failed to stop even after SIGKILL	Listener process could not be terminated even with SIGKILL. Solution: Contact your Oracle Administrator for assistance.
lsnrctl not found in %s/bin	The client utility <code>lsnrctl</code> is not found in the <code>\$ORACLE_HOME/bin</code> directory. Solution: Verify that the Oracle home has been correctly specified and that this executable is present.
lsnrctl operation timed out	The <code>tnslsnr</code> process is not responding. Solution: Verify the underlying network protocol.



Resource Type Definitions



This appendix lists the resource type definitions and attribute definitions of the Oracle and Netlsnr agents.

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 table explains the attributes associated with the agent. The Required attributes table lists the attributes that must be configured for the agent to function properly.

Oracle Resource Type

```
type Oracle (  
    static str SupportedActions [] = {VRTS_GetInstanceName,  
        VRTS_GetRunningServices,DBRestrict,  
        DBUndoRestrict,DBResume,DBSuspend,DBTbspBackup}  
    static str ArgList[] = { Sid, Owner, Home, Pfile, StartUpOpt,  
        ShutDownOpt, EnvFile, AutoEndBkup, DetailMonitor, User, Pword,  
        Table, MonScript, AgentDebug, Encoding }  
    str Sid  
    str Owner  
    str Home  
    str Pfile  
    str StartUpOpt = STARTUP  
    str ShutDownOpt = IMMEDIATE  
    str EnvFile  
    boolean AutoEndBkup = 1  
    int DetailMonitor  
    str MonScript = "./bin/Oracle/SqlTest.pl"  
    str User  
    str Pword  
    str Table  
    boolean AgentDebug = 0  
    str Encoding  
)
```



Attribute Definitions

Required Attributes	Type and Dimension	Definition
Sid	string-scalar	The variable <code>\$ORACLE_SID</code> that represents the Oracle instance. The <code>Sid</code> is considered case-sensitive by the Oracle agent and by the Oracle database server.
Owner	string-scalar	The Oracle user, as the defined owner of executables and database files in <code>/etc/passwd</code> .
Home	string-scalar	The <code>\$ORACLE_HOME</code> path to Oracle binaries and configuration files. For example, you could specify the path as <code>/opt/ora_home</code> . Note Do not append a slash (/) at the end of the path.
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 <code>SPFILE</code> parameter. See the Oracle documentation for more information.

Optional Attributes	Type and Dimension	Definition
StartUpOpt	string-scalar	Startup options for the Oracle instance. This attribute can take the following values: <ul style="list-style-type: none">♦ <code>STARTUP</code>♦ <code>RESTRICTED</code>♦ <code>RECOVERDB</code>♦ <code>SRVCTLSTART</code>♦ <code>CUSTOM</code> Default is <code>STARTUP</code> .

Optional Attributes	Type and Dimension	Definition
ShutDownOpt	string-scalar	<p>Shut down options for the Oracle instance. This attribute can take the following values:</p> <ul style="list-style-type: none"> ♦ IMMEDIATE ♦ TRANSACTIONAL ♦ SRVCTLSTOP ♦ CUSTOM <p>Default is IMMEDIATE</p>
EnvFile	string-scalar	<p>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.</p> <p>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.</p>
AutoEndBkup	integer-scalar	<p>Setting the AutoEndBkup attribute to a non-zero value takes the datafiles in the database out of the backup mode, during Online.</p> <p>Default = 1</p>
DetailMonitor	integer-scalar	<p>Setting this flag to a non-zero enables detail monitoring for Oracle. 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 monitor intervals.</p> <p>Default = 0</p>
MonScript	string-scalar	<p>Pathname to the script provided for detail monitoring. The default (basic monitoring) is to monitor the database PIDs only.</p> <p>The pathname to the supplied detail monitor script is /opt/VRTSvcs/bin/Oracle/SqlTest.pl.</p> <p>MonScript also accepts a pathname relative to \$VCS_HOME. A relative pathname should start with “./”, as in the path ./bin/Oracle/SqlTest.pl.</p>
User	string-scalar	<p>Internal database user. Connects to the database for detail monitoring.</p>



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, described on " Encrypting Passwords " on page 64.
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 Oracle and Solaris documentation for respective encoding values. The default is "".
AgentDebug	boolean-scalar	Additional debug messages are logged when this flag is set. Default = 0

Netlsnr Resource Type

```

type Netlsnr (
    static str SupportedActions [] = {VRTS_GetInstanceName,
        VRTS_GetRunningServices }
    static str ArgList[] = { Owner, Home, TnsAdmin, Listener, EnvFile,
        MonScript, LsnrPwd, AgentDebug, Encoding }
    str Owner
    str Home
    str TnsAdmin
    str Listener
    str EnvFile
    str MonScript = "./bin/Netlsnr/LsnrTest.pl"
    str LsnrPwd
    boolean AgentDebug = 0
    str Encoding
)

```

Attribute Definitions

Required Attributes	Type and Dimension	Definition
Owner	string-scalar	Oracle user, as the defined owner of executables and database files in <code>/etc/passwd</code> .
Home	string-scalar	The <code>\$ORACLE_HOME</code> path to Oracle binaries and configuration files. For example, you could specify the path as <code>/opt/ora_home</code> . Do not append a slash (/) at the end of the path.
TnsAdmin	string-scalar	The <code>\$TNS_ADMIN</code> path to directory in which the Listener configuration file resides (<code>listener.ora</code>). The default is <code>/var/opt/oracle</code> .
Listener	string-scalar	Name of Listener. The default is <code>LISTENER</code> . The name for Listener is considered case-insensitive by the Netlsnr agent and the Oracle database server.



Required Attributes	Type and Dimension	Definition
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, described on “Encrypting Passwords” on page 64.

Optional Attributes	Type and Dimension	Definition
EnvFile	string-scalar	Specifies the full path name of the file that is sourced by the entry point scripts. This file contains the environment variables set by the user for the Oracle listener environment such as LD_LIBRARY_PATH and so on. The syntax for the contents of the file depends on the login shell of Owner. This file must be readable by Owner. The file must not contain any prompts for user input.
MonScript	string-scalar	Pathname to the script provided for detail monitoring. The default (basic monitoring) is to monitor the listener process only. The pathname to the supplied detail monitoring script is \$VCS_HOME/bin/Netlsnr/LsnrTest.pl. MonScript also accepts a pathname relative to \$VCS_HOME. A relative pathname should start with “./”, as in the path ./bin/Netlsnr/LsnrTest.pl.
Encoding	string-scalar	Specifies operating system encoding that corresponds to Oracle encoding for the displayed Oracle output. For example, if Oracle output is in “JAPANESE_JAPAN.JA16EUC,” then “eucJP” is the Solaris value for Encoding. Refer to Oracle and Solaris documentation for respective encoding values. The default is “”.
AgentDebug	boolean	Additional debug messages are logged when this flag is set. Default = 0

Note Detail Monitoring is disabled if the value of the attribute MonScript is invalid or is set to an empty string.

Sample Configurations

This appendix describes typical service groups configured to monitor the state of Oracle in a VCS cluster. 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 *VERITAS Cluster Server Bundled and Enterprise Agents Reference Guide*.

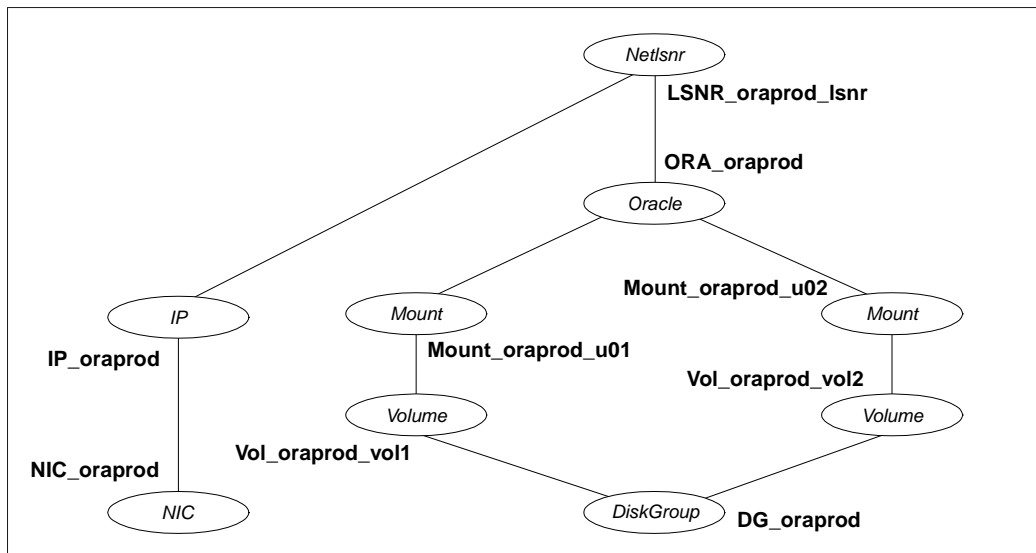
The appendix describes the following configurations

- ◆ “[Single Instance Configuration](#)” on page 86
- ◆ “[Multiple Oracle Instance \(Single Listener\) Configuration](#)” on page 89
- ◆ “[Multiple Instance \(Multiple Listeners\) Configuration](#)” on page 95
- ◆ “[Oracle Configuration With Shared Server Support](#)” on page 100



Single Instance Configuration

The following example describes a typical service group configured to monitor the state of an Oracle instance in a VCS cluster.



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.

If your configuration does not use VERITAS Volume Manager, use the `DiskReservation` resource type to configure shared storage instead of the `DiskGroup` and `Volume` resource types.

Sample Configuration File

```

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

cluster vcs (
)

system vcssun48 (
)

system vcssun49 (

```

```

)

group ORA_PROD_Group (
  SystemList = { vcssun48 = 0, vcssun49 = 1 }
  AutoStartList = { vcssun48 }
)

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

IP IP_oraprod (
  Device = hme0
  Address = "10.212.102.13"
)

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
)

Netlsnr LSNR_oraprod_lsnr (
  Owner = oraprod
  Home = "/prod/u01/oracle/product/8.1.5"
  TnsAdmin = "/prod/u01/oracle/network/admin"
  Listener = LISTENER_PROD
  MonScript = "../bin/Netlsnr/LsnrTest.pl"
)

```

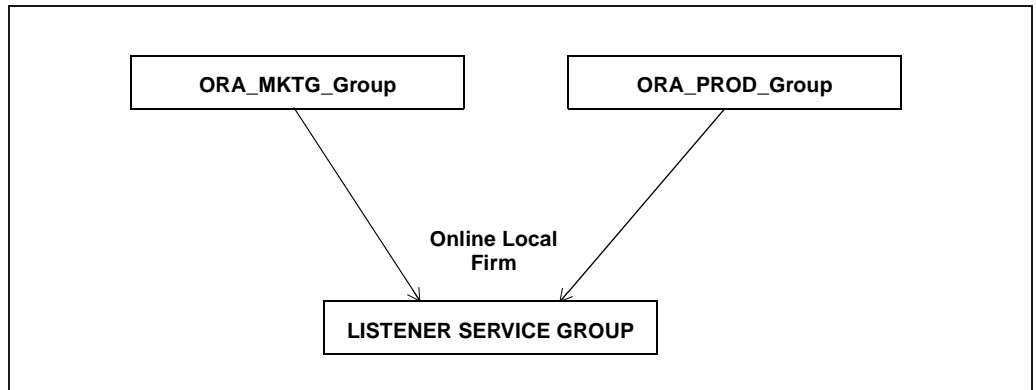


```
Oracle ORA_oraprod (  
  Sid = PROD  
  Owner = oraprod  
  Home = "/prod/u01/oracle/product/8.1.5"  
  Pfile = "/prod/u01/oracle/admin/pfile/initPROD.ora"  
  EnvFile = "/tmp/env.sh"  
  DetailMonitor = 1  
  MonScript = "../bin/Oracle/SqlTest.pl"  
  User = thor  
  Pword = hv1TptWvj  
  Table = thor  
)  
  
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
```


Multiple Oracle Instance (Single Listener) Configuration

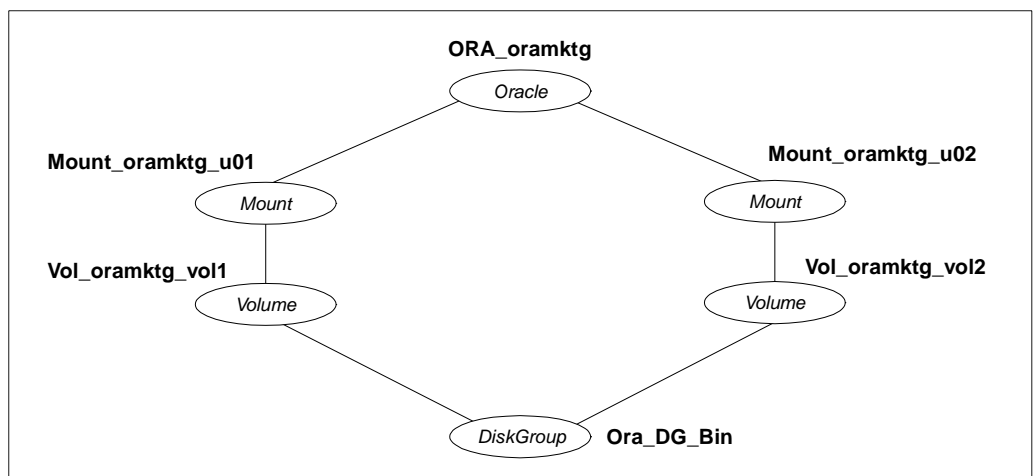
The following example describes a typical VCS configuration to monitor two Oracle instances sharing a listener.

This configuration has a service group for each Oracle instance. The Listener too is configured in a separate service group.

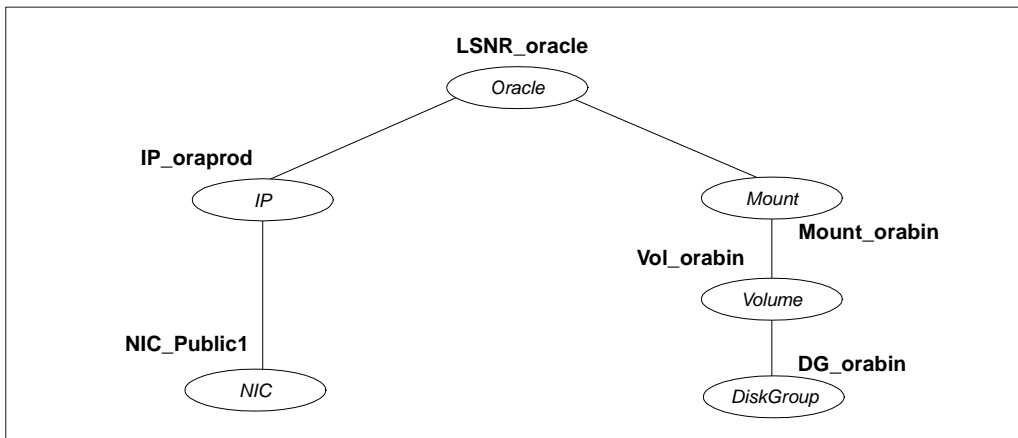


The Oracle service groups are made dependent on the Listener service group using an Online Local Firm dependency.

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.



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.



The Oracle server can be started after the Listener service group and the resources in the Oracle service group are brought online.

If your configuration does not use VERITAS Volume Manager, use the DiskReservation resource type to configure shared storage instead of the DiskGroup and Volume resource types.

Sample Configuration File

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

cluster vcs (
)

system vcssun48 (
)

system vcssun49 (
)

group ORA_MKTG_Group (
    SystemList = { vcssun48 = 0, vcssun49 = 1 }
    AutoStartList = { vcssun48 }
)
```

```
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/u01/oracle/product/8.1.5"  
    Pfile = "/orahome/u01/oracle/admin/pfile/initMKTG.ora"  
    EnvFile = "/tmp/env.sh"  
    DetailMonitor = 1  
    MonScript = "../bin/Oracle/SqlTest.pl"  
    User = thor  
    Pword = hvlTptWvj  
    Table = thor  
)  
  
Volume Vol_oramktg_vol1 (  
    Volume = u01-vol  
    DiskGroup = ora_mktg_dg  
)  
  
Volume Vol_oramktg_vol2 (  
    Volume = u02-vol  
    DiskGroup = ora_mktg_dg  
)
```



```
requires group Parallel_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 = { vcssun48 = 0, vcssun49 = 1 }
  AutoStartList = { vcssun48 }
)

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/u01/oracle/product/8.1.5"
  Pfile = "/orahome/u01/oracle/admin/pfile/initPROD.ora"
  EnvFile = "/tmp/env.sh"
  DetailMonitor = 1
  MonScript = "../bin/Oracle/SqlTest.pl"
  User = thor
  Pword = hvlTptWvj
  Table = thor
)
```



```

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

Volume Vol_oraprod_vol2 (
    Volume = u02-vol
    DiskGroup = ora_prod_dg
)

requires group Parallel_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 Parallel_Service (
    SystemList = { vcssun48 = 0, vcssun49 = 1 }
    Parallel = 1
    AutoStartList = { vcssun48, vcssun49 }
)

DiskGroup DG_orabin (
    DiskGroup = ora_bin_dg
    StartVolumes = 0
    StopVolumes = 0
)

IP IP_oraprod (
    Device = hme0
    Address = "10.212.102.13"
)

Mount Mount_orabin (
    MountPoint = "/orahome/u01"
    BlockDevice = "/dev/vx/dsk/ora_bin_dg/u01-vol"
    FSType = vxfs
    FsckOpt = "-n"
)

NIC NIC_Public1 (
    Device @vcssun48 = hme0
    Device @vcssun49 = hme0
)

```



```
Netlsnr LSNR_oracle (  
  Owner = oracle  
  Home = "/orahome/u01/oracle/product/8.1.5"  
  TnsAdmin = "/orahome/u01/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
```

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 those described in the [“Single Instance Configuration”](#) on page 86.

Sample Configuration File

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

cluster vcs (
)

system vcssun48 (
)

system vcssun49 (
)

group ORA_MKTG_Group (
  SystemList = { vcssun48 = 0, vcssun49 = 1 }
  AutoStartList = { vcssun48 }
)

DiskGroup DG_oramktg (
  DiskGroup = ora_mktg_dg
  StartVolumes = 0
  StopVolumes = 0
)

IP IP_oramktg (
  Device = hme0
  Address = "10.212.102.14"
)

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 = "/mktg/u01/oracle/product/8.1.5"  
  TnsAdmin = "/mktg/u01/oracle/network/admin"  
  Listener = LISTENER_MKTG  
  MonScript = "../bin/Netlsnr/LsnrTest.pl"  
)  
  
Oracle ORA_oramktg (  
  Sid = MKTG  
  Owner = oramktg  
  Home = "/mktg/u01/oracle/product/8.1.5"  
  Pfile = "/mktg/u01/oracle/admin/pfile/initMKTG.ora"  
  EnvFile = "/tmp/env.sh"  
  DetailMonitor = 1  
  MonScript = "../bin/Oracle/SqlTest.pl"  
  User = thor  
  Pword = hvlTptWvj  
  Table = thor  
)  
  
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 = { vcssun48 = 0, vcssun49 = 1 }
  AutoStartList = { vcssun48 }
)

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

IP IP_oraprod (
  Device = hme0
  Address = "10.212.102.13"
)

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"
)
```



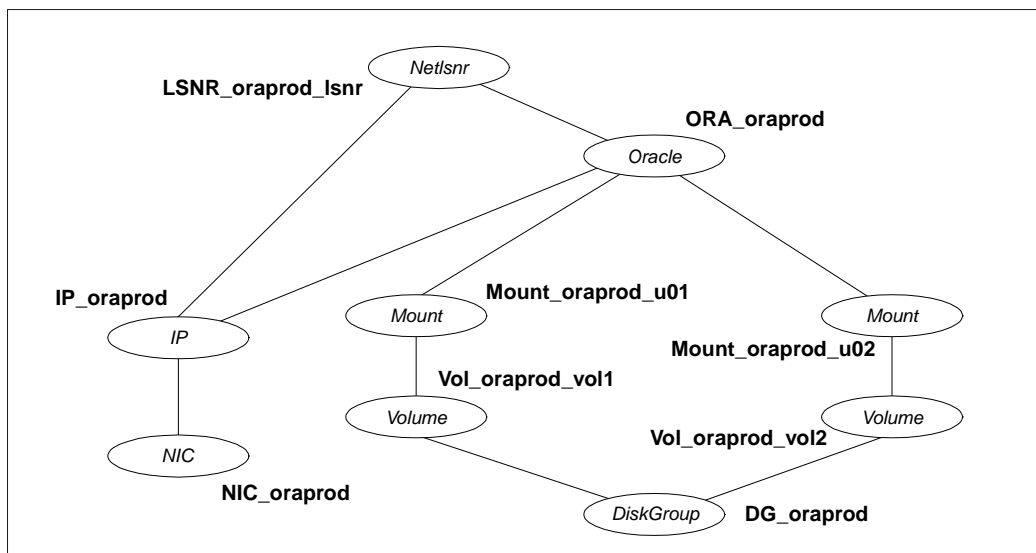
```
Netlsnr LSNR_oraprod_lsnr (  
  Owner = oraprod  
  Home = "/prod/u01/oracle/product/8.1.5"  
  TnsAdmin = "/prod/u01/oracle/network/admin"  
  Listener = LISTENER_PROD  
  MonScript = "../bin/Netlsnr/LsnrTest.pl"  
)  
  
Oracle ORA_oraprod (  
  Sid = PROD  
  Owner = oraprod  
  Home = "/prod/u01/oracle/product/8.1.5"  
  Pfile = "/prod/u01/oracle/admin/pfile/initPROD.ora"  
  EnvFile = "/tmp/env.sh"  
  DetailMonitor = 1  
  MonScript = "../bin/Oracle/SqlTest.pl"  
  User = thor  
  Pword = hvlTptWvj  
  Table = thor  
)  
  
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 = { vcssun48 = 0, vcssun49 = 1 }  
  Parallel = 1  
  AutoStartList = { vcssun48, vcssun49 }  
)  
  
NIC NIC_Public1 (  
  Device @vcssun48 = hme0  
  Device @vcssun49 = hme0  
)
```



Oracle Configuration With Shared Server Support

The following example describes a typical service group configured to monitor Oracle 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.

If your configuration does not use VERITAS Volume Manager, use the `DiskReservation` resource type to configure shared storage instead of the `DiskGroup` and `Volume` resource types.

Sample Configuration File

```

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

cluster vcs (
)

system vcssun48 (
)

system vcssun49 (

```

```

)

group ORA_PROD_Group (
  SystemList = { vcssun48 = 0, vcssun49 = 1 }
  AutoStartList = { vcssun48 }
)

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

IP IP_oraprod (
  Device = hme0
  Address = "10.212.102.13"
)

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
)

Netlsnr LSNR_oraprod_lsnr (
  Owner = oraprod
  Home = "/prod/u01/oracle/product/8.1.5"
  TnsAdmin = "/prod/u01/oracle/network/admin"
  Listener = LISTENER_PROD
  MonScript = "../bin/Netlsnr/LsnrTest.pl"
)

Oracle ORA_oraprod (

```



```
Sid = PROD
Owner = oraprod
Home = "/prod/u01/oracle/product/8.1.5"
Pfile = "/prod/u01/oracle/admin/pfile/initPROD.ora"
EnvFile = "/tmp/env.sh"
DetailMonitor = 1
MonScript = "../bin/Oracle/SqlTest.pl"
User = thor
Pword = hv1TptWvj
Table = thor
)

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
```

Best Practices for Multiple Oracle Instance Configurations



This appendix lists some Best Practices for VCS 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 `/etc/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 several 8.1.x instances and just one listener, set up a parallel service group.
- ◆ If your configuration has different versions of Oracle, create a separate `$ORACLE_HOME` for each Oracle version.
- ◆ Follow the Optimal Flexible Architecture (OFA) standard (`/uwx/<SID>`). In cluster configurations, you could adapt the standard to make it more application-specific. For example, `/app/uwx/<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.



-
- ◆ If you create a single user named oracle and define the variables required by Oracle, you must redefine at minimum the \$ORACLE_HOME and \$ORACLE_SID variables every time you want to invoke `svrmgrl`.

If you define several Oracle users in the `passwd` file, with each user having the appropriate environment variables, so you could easily identify which level of Oracle code you are running.

- ◆ The pfiles must be coordinated between systems. For the same instance of a database, ensure that the pfiles referenced are identical across the nodes.

Using the SPFILE in an Oracle Cluster



Oracle versions prior to Oracle9i used an initialization file `initSID.ora`, a text file, to start database instances. Changes applied to instance parameters during a database session were not saved to the file; you had to manually apply them to the initialization file.

Oracle 9i 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

For versions above Oracle9i, you can start an Oracle instance in the following ways:

- ◆ Using the default SPFILE `spfileSID.ora`
- ◆ Using the default init.ora file `initSID.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 UNIX 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:

- ✓ `SPFILESID.ora`
- ✓ `SPFILE.ora`
- ✓ `initSID.ora`



Using the SPFILE in a VCS Cluster

When using the Oracle enterprise agent, 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 VERITAS recommends that the SPFILE be located on shared storage, preferably at the same location as that of the database control files.

Specifying the SPFILE Location in the PFILE

To specify the location of the SPFILE in a PFILE, create 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/spfileoral.ora
```

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

```
startup pfile=location_of_pfile
```

Creating 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 UNIX).

Index

Symbols

- `$ORACLE_HOME`
 - limitation for 16
 - location 14
 - on local disk 14
 - on shared disk 14

A

- Agent operations
 - Netlsnr agent 3
 - Oracle agent 2
- Attribute definitions
 - Netlsnr agent 83
 - Oracle agent 80
- Attributes
 - editing 61
 - modifying 61

C

- Cluster Manager 59
- Configuration
 - modifying 72
 - typical setup 10
- Configuration files 85
- Configuration wizard 49
- Core files, location of 15

D

- Database tablespaces 15
- Detail Monitoring
 - about 65
 - disabling for Netlsnr 68
 - disabling for Oracle 68
 - enabling for Netlsnr 68
 - enabling for Oracle 68
 - setting up 65

E

- Editing attributes 61
- Encrypting passwords 64

Enterprise agent

- about 1
- configuring using Cluster Manager 59
- configuring using command-line 62
- configuring using wizard 49
- disabling 71
- error messages 73
- installing 33
- modifying configuration 72
- removing 72
- troubleshooting 73
- typical setup 10
- upgrading 35

- Error messages 73

F

- Failover 9

H

- Hot backup 17

M

- Message log 73
- Monitoring options 9

N

- Netlsnr agent
 - attribute definitions 83
 - operations 3
 - resource type 83
- Netlsnr agent attributes
 - AgentDebug 84
 - Encoding 84
 - EnvFile 84
 - Home 83
 - Listener 83
 - LsnrPwd 84
 - MonScript 84
 - Owner 83
 - TnsAdmin 83



NLS information, defining 16

O

Offlining service group 70

Onlining service group 69

Operations

Netlsnr agent 3

Oracle agent 2

Oracle

database tablespaces 15

error handling 6

installing 13

multiple instances 15

shutdown options 5

startup options 4

supported configurations 38

Oracle agent

attribute definitions 80

operations 2

resource type 79

Oracle agent attributes

AgentDebug 82

AutoEndBkup 81

DetailMonitor 81

Encoding 82

EnvFile 81

Home 80

MonScript 81

Owner 80

Pfile 80

Pword 82

ShutDownOpt 81

Sid 80

StartUpOpt 80

Table 82

User 81

Oracle configuration

multiple instances (multiple listeners) 44

multiple instances (single listener) 41

shared server support 46

single instance 39

Oracle database, hot backup 17

P

Passwords, encrypting 64

R

Removing agent 69

Resource attributes, editing 61

Resource type

Netlsnr 83

Oracle 79

S

Sample configuration files 85

Service group

bringing online 69

switching 70

taking offline 70

Shutdown options 5

SPFILE

creating 106

location of 106

using in VCS 105

Startup options 4

Switching service group 70

T

Troubleshooting 73

V

vsencrypt utility 64

W

Wizard 49

