Cluster Server Agent for Oracle Data Guard Installation and Configuration Guide

AIX, Linux, Solaris

6.2

July 2016

VERITAS™

Cluster Server Agent for Oracle Data Guard Installation and Configuration Guide

Agent Version: 6.2

Document version: 6.2 Rev 1

Legal Notice

Veritas Technologies LLC
500 E Middlefield Road
Mountain View, CA 94043

http://www.veritas.com

Technical Support

Technical Support maintains support centers globally. Technical Support's primary role is to respond to specific queries about product features and functionality. The Technical Support group also creates content for our online Knowledge Base. The Technical Support group works collaboratively with the other functional areas within the company to answer your questions in a timely fashion.

Our support offerings include the following:

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

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

www.veritas.com/support

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

Contacting Technical Support

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

www.veritas.com/support

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

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

- Product release level
- Hardware information
- Available memory, disk space, and NIC information

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

Licensing and registration

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

www.veritas.com/support

Customer service

Customer service information is available at the following URL:

www.veritas.com/support

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

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

Support agreement resources

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

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

Contents

Chapter 1

Introducing the agent for Oracle Data Guard

This chapter includes the following topics:

About the agent for Oracle Data Guard

The Cluster Server agent for Oracle Data Guard provides failover support and recovery in an environment that uses Oracle Data Guard to replicate data between Oracle databases.

The agent monitors and manages the state of replicated Oracle databases that run on Cluster Server nodes. The Data Guard resource is online on the system with the primary database server. The agent makes sure that Oracle Data Guard replicates the database information from the primary database server to the standby database server.

The Oracle Data Guard agent supports global and replicated data clusters.

The Cluster Server agent for Oracle Data Guard includes the following two agents:

- The **Oracle Data Guard** agent (resource type OraDG), which uses SQL *Plus for all operations. This agent is supported only in Oracle Single Instance environment.

- The **Oracle Data Guard Broker** agent (resource type OraDGBroker), which uses the Data Guard broker command-line interface DGMGRL. This agent is supported in Oracle Single Instance as well as Oracle RAC environment.

  Note: The Oracle Data Guard Broker agent is not supported on HP-UX platform.

Note: Veritas recommends configuring only one of the above agents to configure Data Gurad replication for a database.

The Cluster Server agent for Data Guard does not support database environments under the control of Oracle Enterprise Manager.

Supported software

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

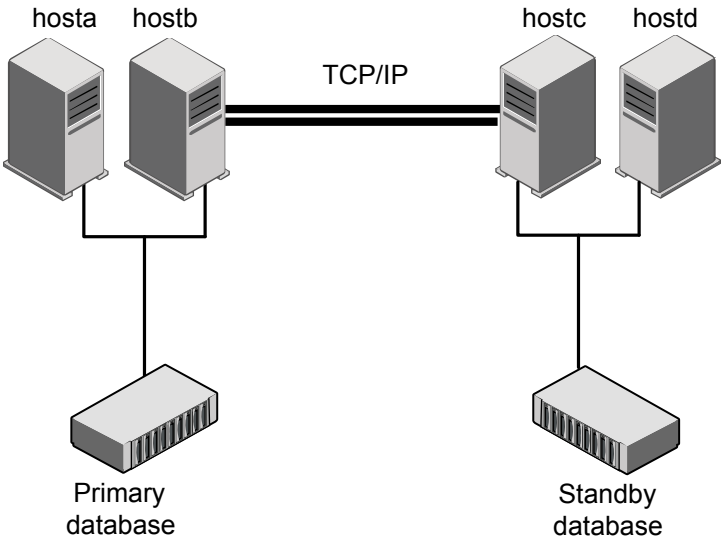
For supported Oracle database versions of Oracle Data Guard agent, refer the support matrix at https://www.veritas.com/support/en_US/article.DOC5081

Note: With Oracle Data Guard Agent, Veritas supports only administrator-managed database (container and pluggable databases are not supported).

Typical Oracle Data Guard setup in a Cluster Server cluster

Figure 1-1 displays a typical cluster setup in a Data Guard environment.

Figure 1-1 Typical clustering setup for the agent



Clustering in a Data Guard environment typically consists of the following hardware infrastructure:

- The primary database instance (db1) sends redo data across a TCP/IP link to a standby database instance (db2). A local cluster protects the primary database and makes it highly available.
- The standby database instance applies the redo information to a physical copy of the primary database.
- The primary and standby sites must be connected through a single TCP/IP network connection. This link can be shared with Cluster Server global clusters for heartbeat communication.
- Network heartbeating between the two data centers to determine their health; this network heartbeating could be LLT or TCP/IP.
  See "About cluster heartbeats" on page 40.

Agent functions for the Data Guard agent

The Oracle Data Guard agent monitors and manages the state of replicated Oracle database that runs on Cluster Server nodes. Agent functions bring resources online,

take them offline, and perform different monitoring actions. Agent functions are also known as entry points.

The agent also supports DataGuard role transition.

See "About DataGuard role transition" on page 15.

Table 1-1 Agent functions

Function	Description
online	Creates a lock file on the local host to indicate that the resource is online. Depending on the role of the database, the agent performs actions to make the database accessible. See "About the Oracle Data Guard agent's online function" on page 14.
offline	Removes the lock file on the local node. Because a switch of the replication direction, promoting the standby and demoting the primary is executed on the target node. Oracle reconfiguration is not done as part of offline. In case of a complete shutdown, an Oracle resource is responsible to close the database.
monitor	Verifies that the lock file exists. ■ If the lock file does not exist, the monitor entry point reports the status of the resource as offline. ■ If the lock file exists, the agent checks if the role of the database is still PRIMARY and the open mode is WRITE.
open	Creates a lock file in the local agent directory if the role of the database is PRIMARY and the open mode is WRITE.
clean	Removes the lock file for the following resource states: ■ OFFLINE TIMEOUT ■ OFFLINE INEFFECTIVE ■ ONLINE TIMEOUT ■ UNEXPECTED OFFLINE ■ MONITOR HUNG
info	Reports the state and the role of the database.
start_stb_curlog.sql	Custom startup script for the Cluster Server agent for Oracle. See "About the custom startup script for the Oracle agent" on page 40.

Table 1-1 Agent functions *(continued)*

Function	Description
actions/DGStatus	Reports the current state and role of the database in real time.
actions/DGDemotePri	Demotes an active PRIMARY to STANDBY database. The agent calls this action as part of the online entry point from a STANDBY database server, when the database role is switched to PRIMARY. The active STANDBY database node drives a DataGuard database server role transition.
actions/activateStandby	It enables the physical standby database to be opened in a read only mode with redo apply from a mounted state. It works only when the state of database is Mounted.
actions/deactivateStandby	Running this action entry point causes the physical standby database to be shutdown and then mounted with redo apply.
actions/flashbackRecover	It is used to convert a failed primary into a standby database using flashback database. After a failover occurs, the original primary database can no longer participate in the Data Guard configuration until it is repaired and established as a standby database in the new configuration. It works only when flashback is enabled at database level. This feature is enabled only when a new attribute Flashback is set to 1, otherwise by default it is 0, which means disabled.
actions/getremoteSCN	It is used only internally by flashbackRecover action entry point to get the SCN for STANDBY_BECAME_PRIMARY_SCN from new primary (remote node) using the `SELECT TO_CHAR (STANDBY_BECAME_PRIMARY_SCN) FROM V$DATABASE` command.
actions/SnapshotOn	It converts the database to snapshot standby and opens the database for read and write operations. It is invoked internally while bringing the OraDGSnap resource online and should not be run manually.
actions/SnapshotOff	It converts snapshot standby database to physical standby. It is invoked internally while bringing the OraDGSnap resource to offline and should not be run manually.

Table 1-1 Agent functions *(continued)*

Function	Description
action/GetCurrentRPO	Fetches the current point-in time RPO in terms of transport lag, apply lag, apply finish time, and estimated startup time. In addition, the agent also displays the current SCN at the primary and standby databases. You must invoke this action function on the DR system. Transport lag: is the measure of the degree to which the transport of redo to the standby database lags behind the generation of redo on the primary database. Apply lag: is the measure of the degree to which the data in a standby database lags behind the data in the primary database. Apply finish time: is an estimate of the time needed to apply all received, but unapplied redo from the primary database. Estimated startup time: is an estimate of the time needed to start and open the standby database. **Note:** The GetCurrentRPO action function displays the Transport lag, Apply lag, Apply finish time, and Estimated startup time in seconds. **Note:** The agent does not store the computed RPO; make a note of the RPO for future reference.

Note: For action entry point functions (activateStandby, deactivateStandby, flashbackRecover, SnapshotOn, and SnapshotOff action functions), it is recommended to increase the values of the ActionTimeout attribute to 300, and MonitorInterval attribute to 600 for OraDG type. Once the action entry point functions get completed, restore the values to its default value. The default value for ActionTimeout is 30 and MonitorInterval is 60. These attributes are supported for Oracle version 11g and above on AIX, Linux, and Solaris operating systems.

About the Oracle Data Guard agent's online function

Scenario: The agent is online on the primary database

The agent determines the role of the database and the type of open mode using the SQL commands:

```
SELECT DATABASE_ROLE from V$DATABASE;
SELECT OPEN_MODE from V$DATABASE
```

If the role of the replicated database is PRIMARY and the open mode is MOUNT, the agent makes the database accessible for clients as follows:

- Alters the database to open mode READ WRITE.
- Creates a lock file on the local host to indicate that the resource is online.

Scenario: The agent is online on the standby database

If the role of the database is PHYSICAL STANDBY, the agent assumes a primary site fault and reconfigures the database as follows:

- The agent first tries to demote a primary database instance by executing the`DGDemotePri` action inside the remote cluster.
- On the Secondary site, the agent changes the mode of the local database from PHYSICAL STANDBY to PRIMARY.
  The agent stops the reception of redo log information using the SQL command:

  ```
  ALTER DATABASE RECOVER MANAGED STANDBY DATABASE CANCEL
  ```

  The agent changes the role of the database using the SQL command:

  ```
  ALTER DATABASE COMMIT TO SWITCHOVER TO PRIMARY
  ```

- The agent then restarts the local database instance to make the changes effective and creates a lock file in the local agent home directory.

Note: For switchover operation, it is recommended to increase the values of the ActionTimeout attribute to 300, and MonitorInterval attribute to 600 for OraDG type. Once the switchover operation gets completed, restore the values to its default value. The default value for ActionTimeout is 30 and MonitorInterval is 60.

About DataGuard role transition

You can switch the service group in which the DataGuard resource resides using the `hagrp -switch` command.

If the agent is OFFLINE on the original primary, the agent removes the lock file.

If the agent is ONLINE on the former standby, the agent executes the following actions:

- Execute action DGDemotePri on the original primary.
- Alter database role from standby to primary.
- Restart Oracle instance on the new primary.

Note: The Oracle dataguard service group on a remote cluster should not be in a frozen state. If the Oracle dataguard service group is in a frozen state, the remote action fails to execute which results in an incomplete switchover operation.

Agent functions for the Data Guard Broker agent

Agent functions bring resources online, take them offline, and perform different monitoring actions. Agent functions are also known as entry points.

Table 1-2 Agent functions

Function	Description
online	Creates a lock file on the local host to indicate that the resource is online. Depending on the role of the database, the agent performs actions to make the database accessible using the `dgmgrl switchover | failover` command.
offline	Removes the lock file on the local node. Oracle reconfiguration is not done as part of offline. In case of a complete shutdown, an Oracle resource is responsible to close the database.
monitor	Verifies that the lock file exists. ■ If the lock file does not exist, the monitor entry point reports the status of the resource as offline. ■ If the lock file exists, the agent checks the role of the database using the `dgmgrl show database` command and reports the status of the resource as online if the the local database server is PRIMARY.
open	Creates a lock file in the local agent directory if the `dgmgrl show database` command reports the role of the database as PRIMARY.
clean	Removes the lock file for the following resource states: ■ OFFLINE TIMEOUT ■ OFFLINE INEFFECTIVE ■ ONLINE TIMEOUT ■ UNEXPECTED OFFLINE ■ MONITOR HUNG

Table 1-2 Agent functions *(continued)*

Function	Description
actions/DGStatus	Returns the output from the `dgmgrl show database` command.
actions/ActRemote	Freezes or flushes a dependent child group which contains a resource of type Oracle for the same SID. The Data Guard Broker starts or stops the database instances outside of the agent framework. As a precaution, the Data Guard Broker agent temporarily freezes any child group on which the service group with the Broker resource depends. Thus the agent avoids VCS to report an unexpected offline. The Oracle Data Guard Broker may restart the instances after a considerable time after the failover is complete. So, the cluster administrator must manually unfreeze any child service group after the Broker completes the replication switchover or failover in the SF Oracle RAC global cluster environment.
actions/FlashbackRecover	Executes the `dgmgrl reinstate` command for the database that needs flashback recovery after disaster recovery. This action function must be executed on the new primary, typically in the following conditions: ■ After a successful failover is completed as part of a disaster recovery operation, and when the old PRIMARY is available now but in the Physical Standby (disabled) role. The FlashbackRecover action function first tries to restart the old primary database in mount mode and then executes the `dgmgrl reinstate` command. If the recovery fails with the Oracle error codes `ORA-01031`, `ORA-16653`, or `ORA-01017`, the action function retries the procedure. Because the agent restarts the old primary database using `dgmgrl` commands that is outside of VCS control, it temporarily freezes any child group that contains an Oracle resource with the same SID. On successful execution of this action function, the role of the old PRIMARY changes from Physical Standby (disabled) to Physical Standby. **Note:** For this action function to work properly, ensure that the FlashBack is enabled at the primary site as well as the Disaster Recovery site.

Table 1-2 Agent functions *(continued)*

Function	Description
action/GetCurrentRPO	Fetches the current point-in time RPO in terms of transport lag, apply lag, apply finish time, and estimated startup time. You must invoke this action function on the DR system. Transport lag: is the measure of the degree to which the transport of redo to the standby database lags behind the generation of redo on the primary database. Apply lag: is the measure of the degree to which the data in a standby database lags behind the data in the primary database. Apply finish time: is an estimate of the time needed to apply all received, but unapplied redo from the primary database. **Note:** The GetCurrentRPO action function displays the Transport lag, Apply lag, Apply finish time, and Estimated startup time in seconds. The agent does not store the computed RPO; make a note of the RPO for future reference.

About the Oracle Data Guard Broker agent's online function in a SF Oracle RAC environment

The agent determines the role of the database using the `dgmgrl` command option `show database <database-name>`.

If the database is already started as PRIMARY, the agent creates the online lock file and exits.

The Oracle Data Guard Broker agent relies on the Oracle DGMGRL command interface to achieve a standby to primary promotion.

The online function always creates an online lock file to enable database monitoring.

If the database role is STANDBY, the online script assumes that a switch of direction or failover of the replication link is requested. The agent does the following:

- On the node where the Oracle database instance is reported as Physical Standby, the agent initiates a promotion from standby to primary using the Data Guard Broker `dgmgrl` command line interface.
- On the nodes where the database instances are in standby mode, the agent loops and monitors the role of the local instance. The Broker command that is run on the apply instance also takes care of the promotion of all the standby

instances. As soon as the agent finds the role as PRIMARY, the function terminates.

- On the apply instance of standby , the online script requests a `dgmgrl failover` if the agent finds the remote cluster state as FAULTED. In any other case, the script assumes that the primary database instance is still active at the remote site, and requests a local database promotion using `dgmgrl switchover`.

The Oracle Data Guard Broker shuts down all other standby instances and all primary instances except one. Except Apply Instance, the Broker restarts all the instances after the failover or switchover transition is complete. As a precaution, the online script requests a temporary freeze for any child service group which contains a resource of type Oracle with the same Sid attribute value. Thus the agent prohibits any VCS interaction with the resources that the Oracle Broker manipulates as part of a switchover or failover.

The online script monitors the output of the `dgmgrl` command and restarts instances if the Broker requests after reconfiguration of the database profiles. For any database shutdown or startup command, the script uses the `dgmgrl` CLI, so you must configure the Oracle Net to support a database start if the Broker is not active.

Refer to the Oracle Data Guard Broker documentation for more information.

Note: The Oracle dataguard broker service group on a remote cluster should not be in a frozen state. If the Oracle datguard broker service group is in a frozen state, the remote action fails to execute which results in an incomplete switchover operation.

Note: For switchover operation it is recommended to increase the values of the ActionTimeout attribute to 300, and MonitorInterval attribute to 600 for OraDG Broker type. Once the switchover operation gets completed, restore the values to its default value. The default value for ActionTimeout is 30 and MonitorInterval is 60.

About the Oracle Data Guard Broker agent's online function in an Oracle single instance environment

If the database role is STANDBY, the online script assumes that a switch of replication direction or failover of replication role is requested.

The agent determines if the Primary database is reachable by using the `dgmgrl` command option `show database`, and then performs the following action:

- If this command returns error codes that indicate that the Primary database is not reachable, the online function requests a `dgmgrl failover`.
- In any other case, the online function requests a `dgmgrl switchover`.

After the switchover transition is complete, Data Guard Broker restarts the databases, if required.

As a precaution, the online script requests a temporary freeze for any child service group that might contain a resource of type Oracle with the same Sid attribute value. This precautionary measure prohibits any VCS interaction with the resources that the Oracle Broker manipulates as part of a switchover or failover operation.

The online script monitors the output of the `dgmgrl` command and restarts instances, if the Broker requests, after reconfiguration of the database profiles. For any database shutdown or startup command, the script uses the `dgmgrl` CLI, so you must configure Oracle Net to support a database start if the Broker is not active.

Refer to the Oracle Data Guard Broker documentation for more information.

Note: For switchover operation it is recommended to increase the values of the ActionTimeout attribute to 300, and MonitorInterval attribute to 600 for OraDG Broker type. Once the switchover operation gets completed, restore the values to its default value. The default value for ActionTimeout is 30 and MonitorInterval is 60.

Chapter 2

Installing and removing the agent for Oracle Data Guard

This chapter includes the following topics:

Before you install the agent for Oracle Data Guard

Before you install the Cluster Server agent for Oracle Data Guard, ensure that you install and configure the VCS on all nodes in the cluster.

Set up replication and the required hardware infrastructure. For information about setting up Oracle RAC environment, refer to the *Storage Foundation for Oracle RAC Configuration and Upgrade Guide*.

See "Typical Oracle Data Guard setup in a Cluster Server cluster" on page 10.

Installing the agent for Oracle Data Guard

You must install the Oracle Data Guard agent on each node in the cluster. In global cluster environments, install the agent on each node in each cluster.

These instructions assume that you have already installed VCS or SF for Oracle RAC.

Note: The VRTScsodg package contains both the Oracle Data Guard agent and the Oracle Data Guard Broker agent.

To install the agent in a VCS environment

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

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

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

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

   AIX `cd1/aix/vcs/replication/Data Guard_agent/`*`agent_version`*`/pkgs/`

   Linux `cd1/linux/generic/vcs/replication/Data Guard_agent/`*`agent_version`*`/rpms/`

   Solaris `cd1/solaris/dist_arch/vcs/replication/Data Guard_agent/`*`agent_version`*`/pkgs/`

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

4. Log in as a superuser.

5. Install the package.

   AIX
   ```
   # installp -ac -d VRTScsodg.rte.bff VRTScsodg.rte
   ```

   Linux
   ```
   # rpm -ihv \
   VRTScsodg-AgentVersion-Linux_GENERIC.noarch.rpm
   ```

   Solaris
   ```
   # pkgadd -d . VRTScsodg
   ```

Note: On successful installation of the agent, if VCS is running, the agent types definition is automatically added to the VCS configuration.

Upgrading the Oracle Data Guard agent

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

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

   ```
   # ps -ef |grep OraDg
   ```

2. If the Oracle Data Guard agent (OraDG) or Data Guard Broker agent (OraDGBroker) is running, perform the following steps:

   - Offline the Data Guard resource

     ```
     hares -offline data_guard_resource_name -sys node_name
     ```

   - Stop the Data Guart agent

     ```
     haagent -stop agent_name -sys sys_name
     ```

3. Uninstall the agent package from all the nodes.

   See "Removing the agent for Oracle Data Guard" on page 23.

4. Install the new agent on all the nodes.

   See "Installing the agent for Oracle Data Guard" on page 21.

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

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

   ```
   # hagrp -unfreeze GroupName -persistent
   ```

7. Start the agent.

   ```
   haagent -start agent_name -sys sys_name
   ```

Removing the agent for Oracle Data Guard

Before you attempt to remove the agent, make sure the application service group is not online.

You must also make sure that the agent operations are stopped on all the nodes. To do this, run the following command:

```
ps -aef|grep OraDG
```

You must remove the Data Guard agent from each node in the cluster.

To remove the agent, type the following command on each node. Answer prompts accordingly:

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

Linux
```
# rpm -e VRTScsodg
```

Solaris
```
# pkgrm VRTScsodg
```

Note: To uninstall the agent IPS package on a Solaris 11 system:

```
# pkg uninstall VRTScsodg
```

Chapter 3

Configuring the agent for Oracle Data Guard

This chapter includes the following topics:

Configuration concepts for the Cluster Server agent for Oracle Data Guard

Review the resource type definition and the attribute definitions for the agents for Oracle Data Guard. The resource type for both the Oracle Data Guard agent and the Oracle Data Guard Broker agent is defined in the OraDGTypes.cf file.

Resource type and attribute definitions for the Data Guard agent

The resource type definition defines the agent in VCS.

Resource type definition for the Data Guard agent on AIX, and Linux is as follows:

```
type OraDG (
        static keylist RegList = { ComputeDRSLA }
        static keylist SupportedActions = { DGStatus, DGDemotePri,
        activateStandby, deactivateStandby, getremoteSCN,
        flashbackRecover, SnapshotOn, SnapshotOff, GetCurrentRPO}
        static int OnlineRetryLimit = 1
        static int OnlineTimeout = 1200
        static int RestartLimit = 1
```

```
        static str ArgList[] = { LinkRes, AgentDebug, Encoding,
        Flashback, ComputeDRSLA }
        str LinkRes
        int ComputeDRSLA = 1
        boolean AgentDebug = 0
        boolean Flashback = 0
        str Encoding
)
```

Resource type definition for the Data Guard agent on Solaris for VCS 5.1 and later:

```
type OraDG (
        static keylist RegList = { ComputeDRSLA }
        static keylist SupportedActions = { DGStatus, DGDemotePri,
        activateStandby, deactivateStandby, getremoteSCN,
        flashbackRecover, SnapshotOn, SnapshotOff, GetCurrentRPO}
        static int OnlineRetryLimit = 1
        static int OnlineTimeout = 1200
        static int RestartLimit = 1
        static str ArgList[] = { LinkRes, AgentDebug, Encoding,
        Flashback, ComputeDRSLA }
        static int ContainerOpts{} = { RunInContainer=1, PassCInfo=0 }
        str LinkRes
        int ComputeDRSLA = 1
        boolean AgentDebug = 0
        boolean Flashback = 0
        str Encoding
)
```

Resource type definition for the Data Guard agent on Solaris for VCS 5.0 and earlier:

```
type OraDG (
        static str ContainerType = Zone
        static keylist RegList = { ComputeDRSLA }
        static keylist SupportedActions = { DGStatus, DGDemotePri,
        activateStandby, deactivateStandby, getremoteSCN,
        flashbackRecover, SnapshotOn, SnapshotOff, GetCurrentRPO }
        static int OnlineRetryLimit = 1
        static int OnlineTimeout = 1200
        static int RestartLimit = 1
        static str ArgList[] = { LinkRes, AgentDebug, Encoding,
        Flashback, ComputeDRSLA }
        str ContainerName
        str LinkRes
```

```
        int ComputeDRSLA = 1
        boolean AgentDebug = 0
        boolean Flashback = 0
        str Encoding
)
```

Attribute definitions for the Oracle Data Guard agent

Review the description of the agent attributes.

Required attributes

You must assign values to the required attributes.

Table 3-1 Required attributes

Attribute	Description
LinkRes	Name of the Oracle resource that manages the replicated database instance. Type-dimension: string-scalar

Optional attributes

Configuring these attributes is optional.

Table 3-2 Optional attributes

Attribute	Description
AgentDebug	Logs additional debug messages when this flag is set. Type-dimension: string-scalar Default: 0

Table 3-2 Optional attributes *(continued)*

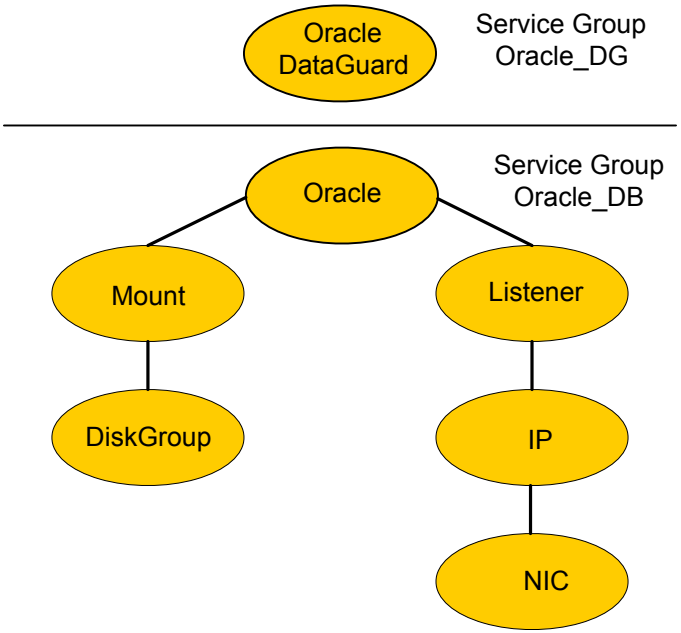
Attribute	Description
Encoding	Specifies the operating system encoding that corresponds to Oracle encoding for the displayed Oracle output. For example, if Oracle output is in “JAPANESE_JAPAN.JA16EUC,” then “eucJP” is the Solaris value for Encoding. Refer to the Oracle and Solaris documentation for respective encoding values. Type-dimension: integer-scalar Default: ““
Flashback	Enables flashback recovery when the failed primary comes up. This attribute is used by the flashbackRecover action entry point. Default: 0
ComputeDRSLA	Enables the computation of Recovery Point Objective (RPO). This attribute cannot be edited. Type-dimension: integer-scalar Default: 1

Sample configuration for the Data Guard agent

Figure 3-1 shows a sample dependency graph.

The Cluster Server service group has a resource of type Data Guard. A second service group contains all necessary resources to control the database instance. The Oracle_DG group depends on the Oracle_DB group, which is an online local soft group dependency.

Figure 3-1 Dependency graph



Note the following variations to a standard Oracle database cluster configuration:

- The Oracle database instance start must be implemented by using a VCS resource of type Oracle with the attribute StartUpOpt set to CUSTOM. See "About the custom startup script for the Oracle agent" on page 40.
- The Oracle resource depends on the Listener resource. The listener process must be already active when the database instance is started because the Data Guard TCP/IP replication links use the Oracle Net Services.
- The IP and NIC resource in the database service group are optional. These resources are only necessary if a cluster on its own protects the primary database. For wide area or site failover, you can implement a transparent network client reconnect.

  To implement a transparent network client reconnect, do one of the following:

  - Use a DNS agent as part of the Data Guard service group
  - Create an alternate Oracle Net Service entries on client machines
- The Oracle resource undergoes an offline-online cycle when promoting a Data Guard standby server to become a primary database. The service group dependency must be soft.

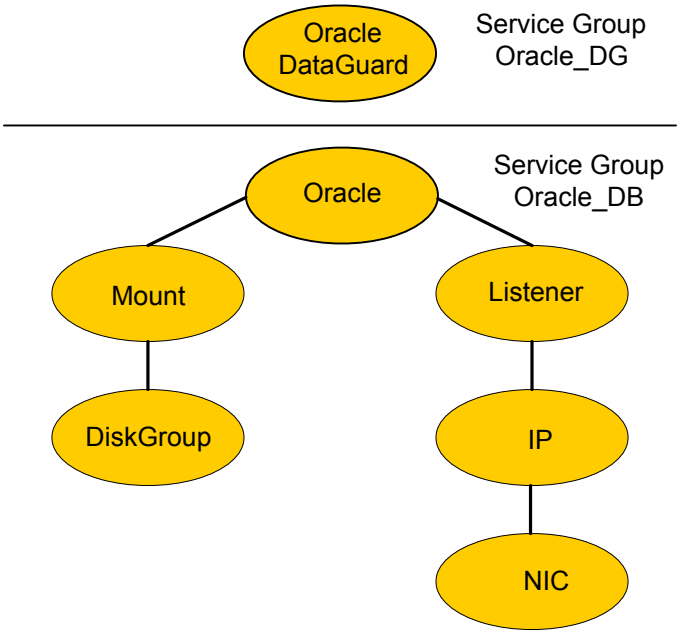
- The name of the Oracle DataGuard resource must be the same in each global cluster configuration. Otherwise, the DemotePri action entry point that is essential for a failover will not work.

Sample configuration in a replicated data cluster environment

Figure 3-2 shows a sample dependency graph.

The Cluster Server service group has a resource of type Data Guard. A second service group contains all necessary resources to control the database instance. The Oracle_DG group depends on the Oracle_DB group, which is an online local soft group dependency.

Figure 3-2 Dependency graph



In a replicated data cluster environment, you can configure a resource of type OraDG in the main.cf file. For more information, See "Sample configuration file for replicated data cluster environment" on page 53.

Resource type and attribute definitions for the Data Guard Broker agent

The resource type definition defines the agent in VCS.

Resource type definition for the Data Guard Broker agent for Solaris, AIX, and Linux is as follows:

```
type OraDGBroker (
    static keylist SupportedActions = { DGStatus, ActRemote,
    FlashbackRecover, GetCurrentRPO}
    static int MonitorTimeout = 120
    static int OnlineRetryLimit = 1
    static int OnlineTimeout = 1800
    static int RestartLimit = 1
    static str ArgList[] = { Sid, DBName, Owner, Home, AgentDebug,
    Encoding, ForceDB, LinkRes, DBUserName, DBPassword }
    static int ContainerOpts{} = { RunInContainer=1, PassCInfo=0 }
    str Sid
    str DBName
    str Owner
    str Home
    boolean AgentDebug = 0
    str Encoding
    str ForceDB
    str LinkRes
    int ComputeDRSLA = 1
    str DBUserName
    str DBPassword
)
```

Attribute definitions for the Oracle Data Guard Broker agent

Review the description of the agent attributes.

Required attributes

You must assign values to the required attributes.

Table 3-3 Required attributes

Attribute	Description
Sid	The Oracle instance identifier. Type-dimension: string-scalar

Table 3-3 Required attributes *(continued)*

Attribute	Description
Owner	The operating system user who is the owner of the Oracle executables. Type-dimension: string-scalar
Home	Location of $ORACLE_HOME where the Oracle binaries are installed. Type-dimension: string-scalar
LinkRes	Name of the Oracle resource that manages the replicated database instance. Type-dimension: string-scalar

Optional attributes

Configuring these attributes is optional.

Table 3-4 Optional attributes

Attribute	Description
DBName	The unique database name. You must configure this attribute in Single Instance environments. Type-dimension: string-scalar **Note:** This attribute is not applicable on HP-UX platform.
DBUserName	Specifies the user name that is used to connect to the Oracle Data Guard Broker. Type-dimension: string-scalar
DBPassword	Specifies the VCS encrypted password that is used to connect to the Oracle Data Guard Broker. Use the `vcsencrypt` command to encrypt the password. Type-dimension: string-scalar
AgentDebug	Logs additional debug messages when this flag is set. Type-dimension: string-scalar Default: 0

Table 3-4 Optional attributes *(continued)*

Attribute	Description
Encoding	Specifies the operating system encoding that corresponds to Oracle encoding for the displayed Oracle output. For example, if Oracle output is in “JAPANESE_JAPAN.JA16EUC,” then “eucJP” is the Solaris value for Encoding. Refer to the Oracle and Solaris documentation for respective encoding values. Type-dimension: integer-scalar Default: ““
ComputeDRSLA	Enables the computation of Recovery Point Objective (RPO). This attribute cannot be edited. Type-dimension: integer-scalar Default: 1

Note: Agent uses DBUserName and DBPassword for DGMGRL login. If DBUserName or DBPassword is not configured, then agent connects to DGMGRL as "/". Veritas recommends to use DBUserName and DBPassword to connect to DGMGRL than "/".

Sample configuration for the Data Guard Broker agent

Figure 3-3 shows a sample dependency graph for the Data Guard Broker agent in an SF Oracle RAC environment.

In an SF Oracle RAC environment, the VCS service group has a resource of type Data Guard Broker. A second service group contains all necessary resources to control the database instance. The Oracle_DGB group depends on the Oracle_DB group, which is an online local soft group dependency.

Figure 3-3 Dependency graph — SF Oracle RAC environment

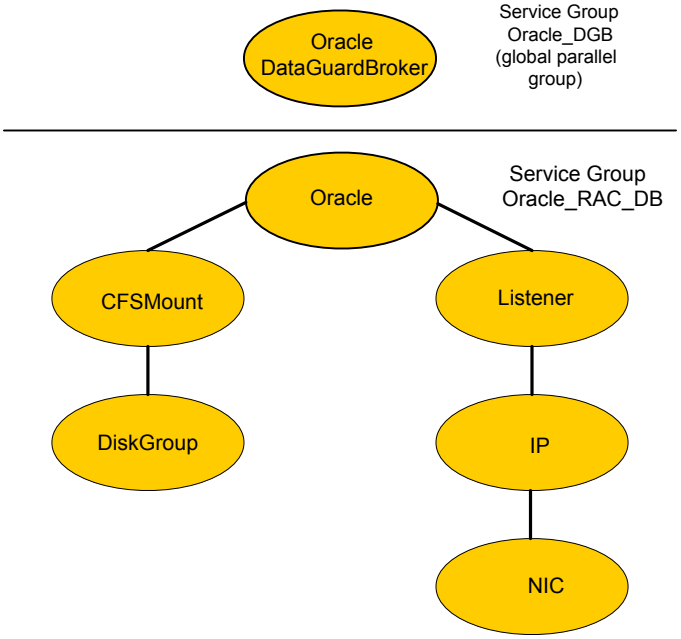
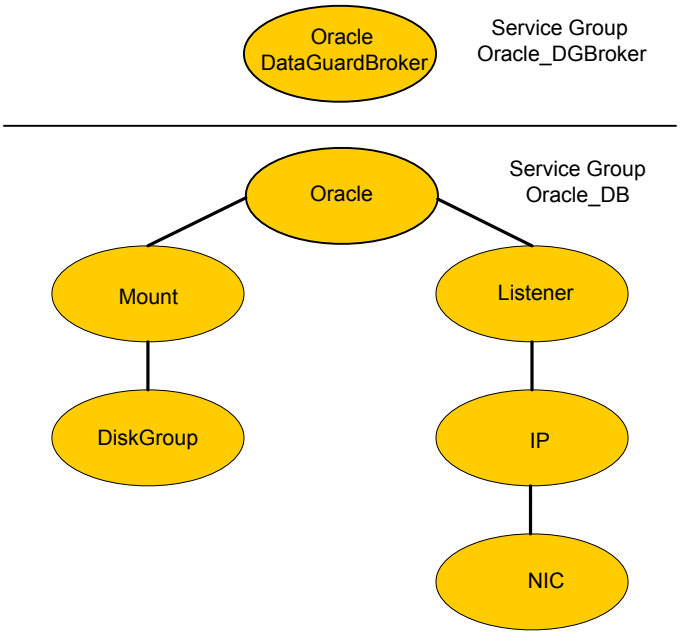


Figure 3-4 shows a sample dependency graph for the Data Guard Broker agent in a single instance Oracle database environment.

In a single instance Oracle database environment, the VCS service group has a resource of type DataGuard Broker. A second service group contains all necessary resources to control the database instance. The Oracle_DGBroker group depends on the Oracle_DB group, which is an online local soft group dependency.

Figure 3-4 Dependency graph — single instance Oracle database environment



You can configure a resource of type OraDGBroker in the main.cf file. For more information, See "Sample configuration file for SF Oracle RAC environment" on page 58.

Note the following variations to a standard Oracle database cluster configuration:

- The Oracle resource or Oracle_RAC_DB service group is optional. The Oracle Data Guard Broker uses its own interface to the database server. The Broker may run in an Oracle Cluster Ready Service (CRS) environment without any assistance from VCS.
- If you have implemented an Oracle resource, the Oracle resource must use StartUpOpt = SRVCTLSTART.
  In an SF Oracle RAC environment, you must configure the Oracle CRS to start the database into "mount" mode.
  See the Oracle Data Guard Broker documentation for Oracle 10g R2.
- You must configure the Oracle network listener to be under the control of the Oracle CRS.

- The name of the Oracle DataGuard Broker resource must be the same in each global cluster configuration. Otherwise, the DemotePri action entry point that is essential for a failover will not work.
- In a single instance environment, if you have implemented an Oracle resource, the Oracle resource must use the StartUpOpt = STARTUP start up option.

Additional concept information for the Data Guard (OraDG) agent

This section provides concept information about the Active Physical Standby, Snapshot Standby, and the Flashback Recover features.

The information in these sections is applicable only to the Data Guard (OraDG) agent; this information is not applicable to the Data Guard Broker (OraDGBroker) agent.

Working of Active Physical Standby feature

The Active Data Guard Option available with Oracle Database 11g Enterprise Edition enables you to open a physical standby database for read-only access for reporting, for simple or complex queries, sorting, or Web-based access while Redo Apply continues to apply changes received from the production database. All queries reading from the physical standby database execute in real time, and return current results. With Active Dataguard, you can offload any operation that requires up-to-date, read-only access to the standby database. To support active standby in Oracle Dataguard agent, we have added two action entry points, activateStandby and deactivateStandby.

activateStandby - On physical standby, it mounts the database in Read-only with Redo apply using below SQL commands:

- `ALTER DATABASE RECOVER MANAGED STANDBY DATABASE CANCEL`
- `ALTER DATABASE OPEN READ ONLY`
- `ALTER DATABASE RECOVER MANAGED STANDBY DATABASE USING CURRENT LOGFILE DISCONNECT`

deactivateStandby - It works differently for different version of Oracle. For 11gR2, only oracle resource needs to offline and then online. But, for 11gR1 it needs to send requests to primary database to send redo log to standby as connection goes down after database offline, using the `ALTER SYSTEM ARCHIVE LOG CURRENT` SQL command.

Working of Snapshot Standby feature

The Snapshot Standby database feature available with Oracle Database 11g Enterprise Edition enables you to open a physical standby database for read-write access when a user requires an updateable snapshot of the physical standby database. A snapshot standby receives and archives redo data from a primary database but does not apply the redo data it receives. The redo data received from the primary database is applied once the snapshot standby database is converted back into a physical standby database, after discarding all local updates to the snapshot standby database. Queries executed on a Snapshot standby database will not provide current results to the user.

To support Snapshot standby in Oracle Dataguard agent, we have added two action entry points:

SnapshotOn - On physical standby, it converts the database to snapshot standby and opens the database for read-write.

- Run `ALTER DATABASE RECOVER MANAGED STANDBY DATABASE CANCEL` to stop redo apply.
- Run `ALTER DATABASE CONVERT TO SNAPSHOT STANDBY` to convert the physical standby to snapshot standby database.
- Run `ALTER DATABASE OPEN` to open the database in read/write mode

SnapshotOff - It converts snapshot standby database to physical standby.

- Takes the database service group offline and then online to get the database in a MOUNTED state.
- Run `ALTER DATABASE CONVERT TO PHYSICAL STANDBY` to convert snapshot database to physical standby.
- Takes the database service group offline.
- Takes the database service group online.

Working of Flashback Recover feature

This feature is used to convert a failed primary into a standby database using flashback database. After a failover occurs, the original primary database can no longer participate in the Data Guard configuration until it is repaired and established as a standby database in the new configuration. To do this, you can use the flashbackRecover action entry point to recover the failed primary database to a point in time before the failover occurred, and then convert it into a physical standby database in the new configuration are completed, the Physical Standby database will become part of the Dataguard configuration and get in sync with the Primary

database server. While executing the action entry point, it would check for the following conditions:

- Value of Flashback attribute of Dataguard resource is set to 1 or 0.
- Flashback is enabled at database level or not.
- Authority- If it is 1, then it would fail as it is running on new primary. Authority is 0, the action entry point would run on the failed primary.
- If failed primary and new primary database's role is PRIMARY, then it would proceed for flashback recovery.

Once the above conditions are met, it would continued as below:

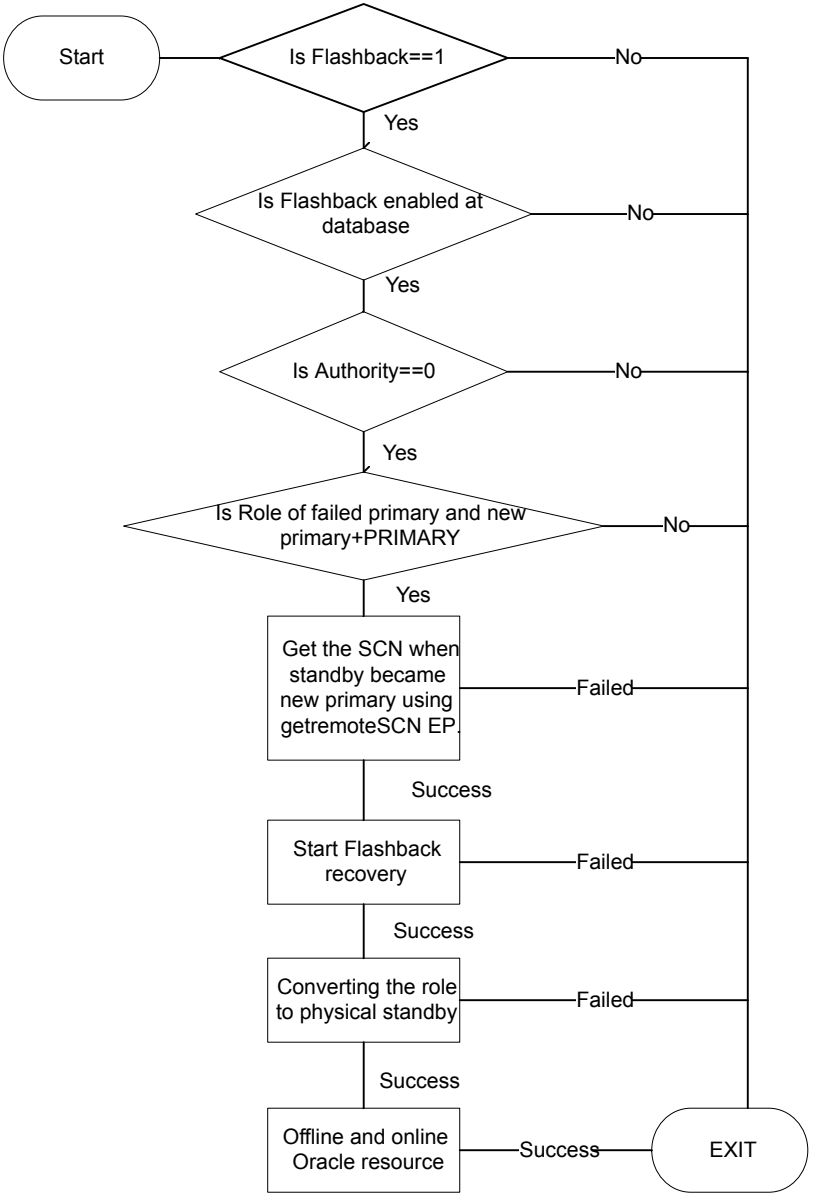
1. Determine the SCN at which the old standby database became the primary database, which is done using the getremoteSCN action entry point.
2. Shutdown the old primary database (if necessary), mount it, and flash it back to the value for STANDBY_BECAME_PRIMARY_SCN that was determined in earlier step.
3. Converts the database to a Physical Standby using the "ALTER DATABASE CONVERT TO PHYSICAL STANDBY" command and then offline and online oracle resource.

After the successful execution of the flashbackRecover action entry point on the failed primary, the new DATABASE_ROLE = PHYSICAL STANDBY and SWITCHOVER_STATUS = SWITCHOVER LATENT or SWITCHOVER PENDING or NOT ALLOWED.

Note: The user or DBA now needs to ensure that the Physical Standby Database receives and applies all the missing changes from the Primary database. Once the manual tasks by the user or DBA are completed, the Physical Standby database will become part of the Dataguard configuration and get in sync with the Primary database server. Do not run the Fire Drill in "ro" or "rw" configuration until the Standby database is brought in sync with the Primary database.

Flowchart for Flashback feature

Figure 3-5 Flowchart for Flashback feature



Before you configure the agent for Oracle Data Guard

Before you configure the agent, review the following information:

- Verify that you have installed the agent on all systems in the cluster.
- Verify the hardware setup for the agent.
  See "Typical Oracle Data Guard setup in a Cluster Server cluster" on page 10.
- Make sure that the cluster has an effective heartbeat mechanism in place.
  See "About cluster heartbeats" on page 40.
- Verify that the clustering infrastructure is in place.
  - If you plan to configure the agent in a global cluster, make sure the global service group for the application is configured.
    For more information, refer to the *Cluster Server Administrator's Guide*.

About cluster heartbeats

In a replicated data cluster, ensure robust heartbeating by using dual, dedicated networks over which the Low Latency Transport (LLT) runs. Additionally, you can configure a low-priority heartbeat across public networks.

In a global cluster, VCS sends ICMP pings over the public network between the two sites for network heartbeating. To minimize the risk of split-brain, VCS sends ICMP pings to highly available IP addresses. VCS global clusters also notify the administrators when the sites cannot communicate.

About preventing split-brain

Split-brain occurs when all heartbeat links between the primary and secondary hosts are cut. In this situation, each side mistakenly assumes that the other side is down. You can minimize the effects of split-brain by ensuring that the cluster heartbeat links pass through a similar physical infrastructure as the replication links. When you ensure that both pass through the same infrastructure, if one breaks, so does the other.

About the custom startup script for the Oracle agent

The Oracle Data Guard agent uses a custom startup script `start_stb_curlog.sql` to start the Oracle agent. The Oracle database instance start has to be implemented by using a Cluster Server resource of type Oracle with the attribute StartUpOpt set to CUSTOM. The necessary file

`/opt/VRTSagents/ha/bin/Oracle/start_custom_<InstID>.sql` can then be implemented as a symbolic link to the `/opt/VRTSvcs/bin/OraDG/start_stb_curlog.sql` file.

Depending on the database role, the agent does the following actions:

- If the database role is PRIMARY, the agent mounts the database and opens it in read-write mode.
- If the database role is PHYSICAL STANDBY, the agent mounts the database. Then, the agent runs the following SQL query to start the replication reception:

```
ALTER DATABASE RECOVER MANAGED STANDBY DATABASE USING
CURRENT LOGFILE DISCONNECT FROM SESSION
```

Note: The Oracle Data Guard agent recommends to set StartUpOpt to CUSTOM for oracle database configured under VCS.

Configuring the agent for Oracle Data Guard

You can configure clustered application in a disaster recovery environment by:

- Changing the database startup profile by adding alternate log destination and creating the necessary Oracle net service entries.
- Creating a second complete database copy on the standby server.
- Adding a new service group with at least the Oracle Data Guard agent. The new service group becomes the parent of the existing Oracle database group.

Refer to the Oracle Data Guard documentation for details on how to configure an Oracle database for Data Guard replication.

After configuration, the application service group must follow the dependency diagram.

Configuring the agent manually in a global cluster

Configuring the agent manually in a global cluster involves the following tasks:

To configure the agent in a global cluster

1. Start Cluster Manager and log on to the cluster.
2. Add a new group with at least one resource of type OraDG or OraDGBroker for VCS or of type OraDGBroker for SFRAC.

3. Configure the attributes of the OraDG or the OraDGBroker resource that you added.
4. Create an online local soft group dependency between the new OraDG or the OraDGBroker group and the existing Oracle database group.
5. Configure the OraDG or the OraDGBroker service group using the Global Group Configuration Wizard as a global group. Refer to the *Cluster Server Administrator's Guide* for more information.
6. Change the ClusterFailOverPolicy from the default, if necessary. Veritas recommends keeping the default, which is Manual, to minimize the chance of failing over on a split-brain.
7. Repeat step 2 through step 6 for each Oracle database service group in each cluster that uses replicated data.

Configuring the agent manually in a replicated data cluster

Configuring the agent manually in a replicated data cluster involves the following tasks:

To configure the agent in a replicated data cluster

1. Start Cluster Manager and log on to the cluster.
2. Create an Oracle service group as hybrid service group. Set the SystemZones attribute of the Oracle service group such that all nodes in the primary RDC zone are in system zone 0 and all nodes in the secondary RDC zone are in zone 1.
3. Create and configure the attributes of the Oracle and related resources within the Oracle service group. Note that some attributes must be localized to reflect values for the hosts that are attached to different database instances.
4. Create an OraDG service group and configure the attributes of the OraDG resource that you added.
5. Create an online local soft group dependency between OraDG and the Oracle database group.

Configuring the agent for Solaris non-global zones

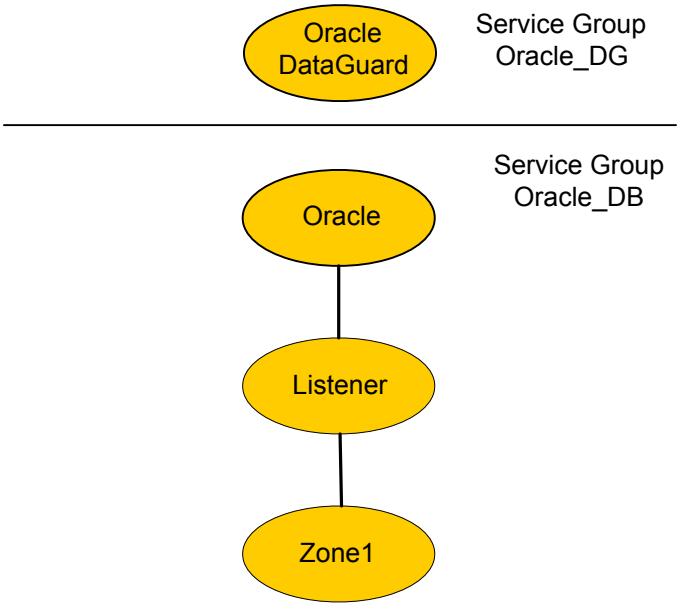
For non-global zone environments (local zones) running under VCS 5.0 or previous versions, you need to add a Zone resource and set up the ContainerName attribute. You must set the ContainerName attribute for the OraDG resource or the OraDGBroker resource, whichever is configured in the production environment. You must also add a Solaris Zone resource under the Listener resource. The Listener

and Oracle resources are executed in the non-global zone and you need to set their ContainerName attribute too.

For non-global zone environments (local zones) running under VCS 5.1, you need to add a Zone resource and set up the ContainerInfo attribute. You must set the ContainerInfo attribute for the OraDG service group or the OraDGBroker service group, whichever is configured in the production environment. You must also add a Solaris Zone resource under the Listener resource. The Listener and Oracle resources are executed in the non-global zone and you need to set the ContainerInfo attribute for the service group containing the oracle and zone resource too.

Figure 3-6 illustrates the dependency tree.

Figure 3-6 Dependency tree



Prepare the configuration with the `hazonesetup` command. This updates the Administrators attribute of the group that operates the Zone, Listener, and the Oracle resource. You need to set the same Administrators attribute for the failover group with OraDG resource manually.

For Oracle Data Guard to work in Zone across GCO you need to follow below steps:

1. On Primary cluster run the `hazonesetup` script which creates a VCS user, Group and updates the Administrators attribute of the group that operates the Zone, Listener, and the Oracle resource.
2. Running the `# hauser -display` command will display the user created and the groups to which privileges are associated.
3. Set the Administrator attribute for the failover group with OraDG resource manually by running the following command:

   ```
   # hauser -addpriv <zone vcs user> Administrator -group <service group name>
   ```

4. Create the same VCS user created by running the `hazonesetup` script on the DR cluster:

   ```
   # hauser -add <zone vcs user> -priv Administrator -group <group with OraDG resource> <group with Zone, Oracle, Listener resource>
   ```

5. Enter password.
6. Repeat step 1 through step 5 and vice versa.
7. Provide the same password as in step 5.

Refer to the *Cluster Server Administrator's Guide* for more information on using Solaris zones.

Configuration example for an OraDG resource:

```
group global_db_rep (
        SystemList = { sec-host = 0 }
                ContainerInfo @ sec-host = { Name = dr_zone , Type = Zone,
                Enabled = 1 }
        ClusterList = { clus-pm = 1, clus-dr = 0 }
        Administrators = { z_zoneres_pm-host, z_zoneres_sec-host }
        )

        OraDG dg_res (
                LinkRes = oradb_stby
                Flashback = 1
                )

        requires group zone_orasg online local soft

group zone_orasg (
```

```
  SystemList = { sec_host = 0 }
        ContainerInfo @ sec-host = { Name = dr_zone , Type = Zone,
        Enabled = 1 }
        Administrators = { z_zoneres_pm-host, z_zoneres_sec-host }
        )

        Netlsnr lsnr (
                Enabled = 0
                Owner = oracle
                Home = "/u01/app/oracle/product/11.2.0/dbhome_1"
                Listener = DGUARD
                )

        Oracle oradb_prod (
                Enabled = 0
                Sid = dguard
                Owner = oracle
                Home = "/u01/app/oracle/product/11.2.0/dbhome_1"
                StartUpOpt = CUSTOM
                )

        Zone zone_res (

                )

        lsnr requires zone_res
        oradb_prod requires lsnr

group fd_sg (
        SystemList = { vcssx208 = 0 }
        ContainerInfo @ sec-host = { Name = dr_zone , Type = Zone,
        Enabled = 1 }
        Administrators = { z_zoneres_pm-host, z_zoneres_sec-host }
        )

        OraDGSnap fd_res (
                Critical = 0
                TargetRes = dg_res
                )
requires group  global_db_rep offline local
```

Configuration example for an OraDGBroker resource:

```
group oradg_broker (
    SystemList = { vcsorat52-04v7 = 0, vcsorat52-03v7 = 1 }
    ContainerInfo = { Name = zone1, Type = Zone, Enabled = 1 }
    ClusterList = { sfcfspri = 0, sfcfssec = 1 }
    Authority = 1
    ClusterFailOverPolicy = Auto
    Administrators = { z_zoneres_vcsorat52-03v8, z_zoneres_vcsorat52-04v7 }
    )

    OraDGBroker dg_broker (
        Sid = racdb
        DBName = racdb
        Owner = oracle
        Home = "/oracle/12c/app/dbbase/dbhome"
        LinkRes = zn_oradb
    )
```

Chapter 4

Managing and testing clustering support for Oracle Data Guard

This chapter includes the following topics:

Failure scenarios for Oracle Data Guard

Review the failure scenarios and agent behavior in response to failure.

All host or all application failure

If all hosts on the primary side are disabled or if the application cannot start successfully on any primary host, the service group fails over.

In replicated data cluster environments, site failure is detected the same way as a total host failure, that is, the loss of all LLT heartbeats. This type of failure is communicated by the VCS engine to the other site.

In global cluster environments, failover requires user confirmation by default. Multiple service groups can fail over in parallel.

Replication link failure

Data Guard detects link failures, monitors the archive logs created on the active primary. When the standby server reconnects to the primary database server, the Data Guard resynchronizes the standby database with all the archive logs. The agent resynchronizes the archive logs since the time of the link failure.

The standby database may not contain the most recent data in the following conditions:

- A failover is initiated due to a disaster at the primary site, and
- A synchronization was in progress

However, the agent is able to execute a role transition from standby to primary. The database contents at the standby site are always consistent.

After recovery of the replication link, the two replicated databases can be logically inconsistent. The database transactions can result in inconsistency in the following conditions:

- The transactions are committed on the original primary after the link failure, and
- The transactions are never replicated to the standby at the time of takeover on the original primary after the link failure

You can get both sites back into a consistent state only if Oracle flash recovery was enabled at both primary and standby database servers. Otherwise, a restart from the last consistent backup can be necessary.

Split-brain in a Data Guard environment

When split-brain occurs in a replicated database cluster, VCS assumes a total disaster because the primary database is unreachable. VCS attempts to start the application. Once the heartbeats are restored, VCS stops the applications on one side and restarts the VCS engine (HAD). This action eliminates concurrency violation of the same group being online at two places simultaneously.

You must resynchronize the databases manually either by using flashback information or the archive logs. Similar to a replication link failure, a complete restart from a backup copy might be necessary.

In a global cluster, you can confirm the failure before failing over the service groups. You can check with the site administrator to identify the cause of the failure. However, when the heartbeat is restored, VCS does not stop HAD at either site. VCS forces you to choose which group to take offline. You must resynchronize the data manually.

Chapter 5

Setting up a fire drill

This chapter includes the following topics:

About fire drills

A fire drill procedure verifies the fault-readiness of a disaster recovery configuration. This procedure is done without stopping the application at the primary site and disrupting user access.

A fire drill is performed at the secondary site using a special service group for fire drills. The fire drill service group uses a read-only copy or snapshot of the database that is used by the application service group. Bringing the fire drill service group online demonstrates the ability of the application service group to come online as a Primary database at the remote site when a failover occurs.

The agent supports Fire drill in a single instance environment for Oracle11gR1 and later.

About the OraDGSnap agent

The OraDGSnap agent is the fire drill agent for Oracle DataGuard. The agent handles how the Standby database can be opened so that users can check the integrity of the database or use it for additional purposes such a reporting, backups, and so on. The behavior of the OraDGSnap agent is based on the SnapMode attribute. If the value is “ro”, the agent will open the Standby database in read only

mode with redo apply. If the SnapMode is set to “rw”, the agent will open the Standby database in read/write mode. In this case, redo logs are received and archived but not applied to the standby database.

Note: OraDGSnap is supported only if the OraDG agent is configured. OraDGSnap is not supported if configured with the OraDGBroker agent.

Note: No changes made on the primary database are visible on the snapshot standby database during the period of the fire drill.

Switchover/Failover of the OraDG resource will fail if the Standby database is in a Snapshot Standby database mode and ClusterFailover policy is set to manual.

OraDGSnap agent functions

The OraDGSnap agent performs the following functions:

Table 5-1 Agent functions

Function	Description
online	It acts based on the value of SnapMode. It invokes activateStandby if it is ‘ro’ and SnapshotOn if ‘rw’ and creates a lock file.
offline	It acts based on the value of SnapMode. It invokes deactivateStandby if it is ‘ro’ and SnapshotOff if ‘rw’ and removes the lock file.
monitor	Verifies the existence of the lock file to make sure the resource is online.
clean	It invokes SnapshotOff or deactivateStandby and removes Lock file.

Resource type definition for the OraDGSnap agent

Following is the resource type definition for the OraDGSnap agent:

```
type OraDGSnap (
        static int OnlineRetryLimit = 1
        static int OnlineTimeout = 1200
        static int RestartLimit = 1
```

```
        static boolean FireDrill = 1
        static str ArgList[] = { TargetRes, SnapMode }
        str TargetRes
        str SnapMode
)
```

Attribute definitions for the OraDGSnap agent

To customize the behavior of the OraDGSnap agent, configure the following attributes:

Table 5-2 Agent attributes

Attribute	Description
TargetRes	Set this attribute to the name of the OraDG type resource Type-Dimension: string-scalar
SnapMode	Specifies whether the Standby database will be open in an Active Standby or Snapshot Standby mode. For Active Standby set this attribute to “ro” and for Snapshot Standby set this attribute to “rw”. Type-Dimension: string-scalar

Before you configure the fire drill service group

Before you configure the fire drill service group, ensure that the following pre-requisites are met:

- Make sure the application service group is configured with a OraDG resource.
- To use the Fire Drill option with SnapMode set to “rw”, the standby database must have Flashback option enabled on the database level.

Sample configuration for a fire drill service group

This is a sample configuration of a fire drill service group where a OraDG resource dgu1 is configured under a global service group and a OraDGSnap resource is configured under the oradgfd fire drill service group. A local offline dependency is created between the fire drill service group and the global service group.

You can configure a resource of type OraDGSnap in the main.cf file as follows:

```
group globaldg (
          SystemList = { vcssx208 = 0 }
          ClusterList = { clus-dg1 = 1, clus-dg2 = 0 }
          )

          OraDG dgu1 (
                  LinkRes @vcssx208 = ora1
                  AgentDebug @vcssx208 = 1
                  Flashback = 1
                 )

         requires group oradgst online local soft


         group oradgfd (
          SystemList = { vcssx208 = 0 }
         )

         OraDGSnap oradgsnap (
             TargetRes = dgu1
                 SnapMode = rw
         )

        requires group globaldg offline local
```

Appendix A

Sample Configurations

This appendix includes the following topics:

About the sample configuration for the agent for Oracle Data Guard

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

Sample configuration file for replicated data cluster environment

In a replicated data cluster environment, you can configure a resource of type OraDG in the main.cf file:

```
group oradg (
 SystemList = { vcssx170 = 0, vcssx171 = 1, vcssx172 = 2, vcssx173 = 3 }
 )

 OraDG oradg (
```

```
 LinkRes = dbresprim
 Flashback = 1
 )

requires group primoragrp online local soft

group primoragrp (
SystemList = { vcssx170 = 0, vcssx171 = 1, vcssx172 = 2, vcssx173 = 3 }
Parallel = 2
SystemZones = { vcssx170 = 1, vcssx171 = 1, vcssx172 = 2, vcssx173 = 2 }
)

DiskGroup dbdgprim (
 DiskGroup @vcssx170 = primdg
 DiskGroup @vcssx171 = primdg
 DiskGroup @vcssx172 = secdg
 DiskGroup @vcssx173 = secdg
 )

 IP lsnrip (
 Device = e1000g0
 Address @vcssx170 = "10.209.81.128"
 Address @vcssx171 = "10.209.81.128"
 Address @vcssx172 = "10.209.81.129"
 Address @vcssx173 = "10.209.81.129"
 NetMask = "255.255.252.0"
 )

Mount dbmntprim (
 MountPoint = "/db"
 BlockDevice @vcssx170 = "/dev/vx/dsk/primdg/primvol"
 BlockDevice @vcssx171 = "/dev/vx/dsk/primdg/primvol"
 BlockDevice @vcssx172 = "/dev/vx/dsk/secdg/secvol"
 BlockDevice @vcssx173 = "/dev/vx/dsk/secdg/secvol"
 FSType = vxfs
 MountOpt = rw
 FsckOpt = "-y"
 )

Mount flashmntprim (
 MountPoint = "/flash"
 BlockDevice @vcssx170 = "/dev/vx/dsk/primdg/flashvol"
 BlockDevice @vcssx171 = "/dev/vx/dsk/primdg/flashvol"
```

```
 BlockDevice @vcssx172 = "/dev/vx/dsk/secdg/flashvol"
 BlockDevice @vcssx173 = "/dev/vx/dsk/secdg/flashvol"
 FSType = vxfs
 MountOpt = rw
 FsckOpt = "-y"
 )

Mount primarcres (
 MountPoint = "/arch"
 BlockDevice @vcssx170 = "/dev/vx/dsk/primdg/archvol"
 BlockDevice @vcssx171 = "/dev/vx/dsk/primdg/archvol"
 BlockDevice @vcssx172 = "/dev/vx/dsk/secdg/archvol"
 BlockDevice @vcssx173 = "/dev/vx/dsk/secdg/archvol"
 FSType = vxfs
 MountOpt = rw
 FsckOpt = "-y"
 )

NIC nicprim (
 Device = e1000g0
 )

Netlsnr lsnr (
 Owner = oracle
 Home = "/u01/app/dbhome"
 EnvFile = "/oracle/oracle/.profile"
 )

Oracle dbresprim (
 Sid @vcssx170 = dguard
 Sid @vcssx171 = dguard
 Sid @vcssx172 = dguardst
 Sid @vcssx173 = dguardst
 Owner = oracle
 Home = "/u01/app/dbhome"
 StartUpOpt = CUSTOM
 )

Volume archvolprim (
 Volume = archvol
 DiskGroup @vcssx170 = primdg
 DiskGroup @vcssx171 = primdg
 DiskGroup @vcssx172 = secdg
```

```
 DiskGroup @vcssx173 = secdg
 )

Volume dbvolprim (
 Volume @vcssx170 = primvol
 Volume @vcssx171 = primvol
 Volume @vcssx172 = secvol
 Volume @vcssx173 = secvol
 DiskGroup @vcssx170 = primdg
 DiskGroup @vcssx171 = primdg
 DiskGroup @vcssx172 = secdg
 DiskGroup @vcssx173 = secdg
 )

Volume flashvolprim (
 Volume = flashvol
 DiskGroup @vcssx170 = primdg
 DiskGroup @vcssx171 = primdg
 DiskGroup @vcssx172 = secdg
 DiskGroup @vcssx173 = secdg
 )

archvolprim requires dbdgprim
dbmntprim requires dbvolprim
dbresprim requires dbmntprim
dbresprim requires flashmntprim
dbresprim requires lsnr
dbresprim requires primarcres
dbvolprim requires dbdgprim
flashmntprim requires flashvolprim
flashvolprim requires dbdgprim
lsnr requires lsnrip
lsnrip requires nicprim
primarcres requires archvolprim
```

Sample configuration file for global cluster environment

In a global cluster environment, you can configure a resource of type OraDG in the main.cf file:

```
group global_db_rep (
        SystemList = { primary-sys1 = 0, primary-sys2 = 1 }
        ClusterList = { dgclus1 = 0, dgclus2 = 1 }
                )

        OraDG dg_res (
                LinkRes = ora_db_prod
                Flashback = 1
                )

        requires group oradb_prod_SG online local soft




group oradb_prod_SG (
        SystemList = { primary-sys1 = 0, primary-sys2 = 1 }
        )

               IP lsnr_ip (
                Device = eth0
                Address = "10.209.71.181"
                NetMask = "255.255.252.0"
                )

        LVMLogicalVolume ora_vol (
                LogicalVolume = OraData
                VolumeGroup = VolGroup01
                )

        LVMVolumeGroup ora_grp (
                VolumeGroup = VolGroup01
                )
       Mount ora_mnt (
                MountPoint = "/u01"
                BlockDevice = "/dev/mapper/VolGroup01-OraData"
                FSType = ext3
                FsckOpt = "-y"
                )

        NIC lsnr_nic (
                Device = eth0
                )
```

```
        Netlsnr ora_db_lsnr (
                Owner = oracle
                Home = "/u01/app/oracle/product/11.2.0/db_1"
                Listener = DGUARD
                )

        Oracle ora_db_prod (
                Sid = dguard
                Owner = oracle
                Home = "/u01/app/oracle/product/11.2.0/db_1"
                StartUpOpt = CUSTOM
                )

        lsnr_ip requires lsnr_nic
        ora_db_lsnr requires lsnr_ip
        ora_db_prod requires ora_mnt
        ora_vol requires ora_grp
        ora_mnt requires ora_vol
        ora_db_prod requires ora_db_lsnr
```

Sample configuration file for SF Oracle RAC environment

A sample main.cf file in an SF Oracle RAC environment is as follows:

```
group oradgb (
        SystemList = { racqa14-1 = 0, racqa14-2 = 1, racqa14-3 = 2,
                 racqa14-4 = 3 }
        ClusterList = { certclust = 0, dsm_aix = 1 }
        Authority = 1
        AutoStartList = { racqa14-1, racqa14-2, racqa14-3, racqa14-4 }
        )

        OraDGBroker dgb (
                Critical = 0
                Sid @racqa14-1 = pri1
                Sid @racqa14-2 = pri2
                Sid @racqa14-3 = pri3
                Sid @racqa14-4 = pri4
                DBName = pri
                Owner = oracle
```

```
                Home = "/oracle/11g/dbbase/dbhome"
                AgentDebug = 1
                LinkRes = pridb
                DBUserName = "system"
                DBPassword = FTLrITi
                )


group oragrp (
        SystemList = { racqa14-4 = 0, racqa14-3 = 1, racqa14-2 = 2,
                 racqa14-1 = 3 }
        Parallel = 1
        AutoStartList = { racqa14-4, racqa14-3, racqa14-2, racqa14-1 }
        )

        CFSMount arch_mnt_1 (
                Critical = 0
                BlockDevice = "/dev/vx/dsk/archdg/vol1"
                MountPoint = "/arch"
                )

        CFSMount datavol_mnt_1 (
                Critical = 0
                BlockDevice = "/dev/vx/dsk/oradg_14/datavol"
                MountPoint = "/oradata"
                )

        CVMVolDg archvol_voldg_1 (
                Critical = 0
                CVMVolume = { vol1 }
                CVMActivation = sw
                CVMDiskGroup = archdg
                )

        CVMVolDg datavol_voldg_1 (
                Critical = 0
                CVMVolume = { datavol }
                CVMVolumeIoTest = { datavol }
                CVMActivation = sw
                CVMDiskGroup = oradg_14
                )

        Oracle pridb (
```

```
        Critical = 0
        Owner = oracle
        DBName = pri
        Home = "/oracle/11g/dbbase/dbhome"
        StartUpOpt = SRVCTLSTART
        Sid @racqa14-4 = pri4
        Sid @racqa14-3 = pri3
        Sid @racqa14-2 = pri2
        Sid @racqa14-1 = pri1
        )

    requires group cvm online local firm
    arch_mnt_1 requires archvol_voldg_1
    archvol_voldg_1 requires datavol_voldg_1
    datavol_mnt_1 requires datavol_voldg_1
    pridb requires arch_mnt_1
    pridb requires datavol_mnt_1
```

Sample configuration file for single instance environment

A sample main.cf file in a single instance environment is as follows:

```
group oradg_broker (
        SystemList = { dblxx64-13-v1 = 0, dblxx64-13-v2 = 1 }
        ClusterList = { dgclus = 0, dgclus1 = 1 }
        Authority = 1
        )

        OraDGBroker dg_broker (
                Sid = sfaedb
                Owner = oragrid
                Home = "/ora_base/db_home"
                AgentDebug = 1
                LinkRes = dbresprim
                DBName = sfaedb
                DBUserName = "system"
                DBPassword = FTLrITi
                )

        requires group primoragrp online local soft
```

```
group primoragrp (
        SystemList = { dblxx64-13-v1 = 0, dblxx64-13-v2 = 1 }
        AutoStartList = { dblxx64-13-v1, dblxx64-13-v2 }
        )

        DiskGroup dbdgprim (
                Critical = 0
                DiskGroup @dblxx64-13-v1 = datadg
                DiskGroup @dblxx64-13-v2 = datadg
                )

        DiskGroup flashdgprim (
                Critical = 0
                DiskGroup @dblxx64-13-v1 = flashdg
                DiskGroup @dblxx64-13-v2 = flashdg
                )

        Mount dbmntprim (
                Critical = 0
                MountPoint = "/data"
                BlockDevice @dblxx64-13-v1 = "/dev/vx/dsk/datadg/datavol"
                BlockDevice @dblxx64-13-v2 = "/dev/vx/dsk/datadg/datavol"
                FSType = vxfs
                MountOpt = rw
                FsckOpt = "-y"
                )

        Mount flashmntprim (
                Critical = 0
                MountPoint = "/flashrec"
                BlockDevice @dblxx64-13-v1 = "/dev/vx/dsk/flashdg/flashvol"
                BlockDevice @dblxx64-13-v2 = "/dev/vx/dsk/flashdg/flashvol"
                FSType = vxfs
                MountOpt = rw
                FsckOpt = "-y"
                )

        Mount primarcres (
                Critical = 0
                MountPoint = "/arch"
                BlockDevice @dblxx64-13-v1 = "/dev/vx/dsk/datadg/archvol"
                BlockDevice @dblxx64-13-v2 = "/dev/vx/dsk/datadg/archvol"
```

```
        FSType = vxfs
        MountOpt = rw
        FsckOpt = "-y"
        )

NIC nicprim (
        Device = eth0
        )

Oracle dbresprim (
        Sid @dblxx64-13-v1 = sfaedb
        Sid @dblxx64-13-v2 = sfaedb
        Owner = oragrid
        Home = "/ora_base/db_home"
        StartUpOpt = STARTUP
        )

Volume archvolprim (
        Critical = 0
        DiskGroup @dblxx64-13-v1 = datadg
        DiskGroup @dblxx64-13-v2 = datadg
        Volume @dblxx64-13-v1 = archvol
        Volume @dblxx64-13-v2 = archvol
        )

Volume dbvolprim (
        Critical = 0
        DiskGroup @dblxx64-13-v1 = datadg
        DiskGroup @dblxx64-13-v2 = datadg
        Volume @dblxx64-13-v1 = datavol
        Volume @dblxx64-13-v2 = datavol
        )

Volume flashvolprim (
        Critical = 0
        DiskGroup @dblxx64-13-v1 = flashdg
        DiskGroup @dblxx64-13-v2 = flashdg
        Volume @dblxx64-13-v1 = flashvol
        Volume @dblxx64-13-v2 = flashvol
        )

archvolprim requires dbdgprim
dbmntprim requires dbvolprim
```

```
dbresprim requires dbmntprim
dbresprim requires flashmntprim
dbresprim requires primarcres
dbvolprim requires dbdgprim
flashmntprim requires flashvolprim
flashvolprim requires flashdgprim
primarcres requires archvolprim
```

Index

R

S

T

U