Oracle® Solaris Cluster Data Service for Oracle Database Guide

Part No: E39654
September 2014, E39654–02

ORACLE

Contents

Figures

Tables

Examples

Using This Documentation

- **Overview** – Provides procedures to install and configure the Oracle Solaris Cluster HA for Oracle Database data service.
- **Audience** – Experienced system administrators with extensive knowledge of Oracle software and hardware.
- **Required knowledge** – Knowledge of the Oracle Solaris operating system, of Oracle Solaris Cluster software, and expertise with the volume manager software that is used with Oracle Solaris Cluster software.

This document is not to be used as a planning or presales guide.

Product Documentation Library

Late-breaking information and known issues for this product are included in the documentation library at http://www.oracle.com/pls/topic/lookup?ctx=E39579.

Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program web site at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Feedback

Provide feedback about this documentation at http://www.oracle.com/goto/docfeedback.

♦ ♦ ♦ CHAPTER 1

Installing and Configuring HA for Oracle Database

This chapter explains how to install and configure Oracle Solaris Cluster HA for Oracle Database (HA for Oracle Database).

This chapter contains the following sections.

Note - You can use Oracle Solaris Cluster Manager to configure this data service. In the Tasks panel of Oracle Solaris Cluster Manager, click Oracle Database to start the configuration wizard. For log-in instructions, see Chapter 13, "Using the Oracle Solaris Cluster GUI," in "Oracle Solaris Cluster System Administration Guide ".

Overview of the Installation and Configuration Process for HA for Oracle Database

The following task maps summarize the tasks for installing and configuring HA for Oracle Database. The tables also provide cross-references to detailed instructions for performing the tasks.

- Table 1-1
- Table 1-2

Perform these tasks in the order that they are listed. If you are using HA for Oracle Database with Oracle Data Guard, perform these tasks on each cluster where your Oracle Database instances are running.

Note - Single instance Oracle ASM is not supported in an Oracle Solaris Cluster 4.2 configuration.

TABLE 1-1 Task Map: Installing and Configuring HA for Oracle Database

Task	Cross-Reference
Plan the HA for Oracle Database installation and configuration	"Planning the HA for Oracle Database Installation and Configuration" on page 17
Prepare the Oracle Solaris Cluster nodes and disks	"Preparing the Oracle Solaris Cluster Nodes and Disks" on page 19
Install the Oracle Database software	"How to Install Oracle Database Software" on page 28
Verify the Oracle Database installation	"How to Verify Oracle Database Installation" on page 29
Create an Oracle database	"How to Create a Primary Oracle Database" on page 30
Set up Oracle Database permissions	"How to Set Up Oracle Database Permissions" on page 31
Install the HA for Oracle Database packages	"Installing the HA for Oracle Database Package" on page 34
Register and configure HA for Oracle Database	"How to Register and Configure HA for Oracle Database Without Oracle Grid Infrastructure (CLI)" on page 41
Verify the HA for Oracle Database installation	"Verifying the HA for Oracle Database Installation" on page 58
Tune the HA for Oracle Database fault monitor	"Tuning the HA for Oracle Database Fault Monitors" on page 60
(Optional) Customize the HA for Oracle Database server fault monitor	"Customizing the HA for Oracle Database Server Fault Monitor" on page 65
(Optional) Upgrade HA for Oracle Database resource types	"Upgrading HA for Oracle Database Resource Types" on page 75
(Optional) Change the role of an Oracle Data Guard instance	"Changing the Role of an Oracle Data Guard Instance" on page 79

TABLE 1-2 Task Map: Installing and Configuring HA for Oracle Database with Clustered Oracle ASM

Task	Cross-Reference
Plan the HA for Oracle Database installation and configuration	"Planning the HA for Oracle Database Installation and Configuration" on page 17
Prepare the Oracle Solaris Cluster nodes and disks	"Preparing the Oracle Solaris Cluster Nodes and Disks" on page 19
Install Oracle Grid Infrastructure software	"Installing Oracle Grid Infrastructure Software" on page 27
Install Oracle Database software	"How to Install Oracle Database Software" on page 28
Verify Oracle Database installation	"How to Verify Oracle Database Installation" on page 29
Create an Oracle database	"How to Create a Primary Oracle Database" on page 30
Set up Oracle Database permissions	"How to Set Up Oracle Database Permissions" on page 31
Install the HA for Oracle Database packages	"Installing the HA for Oracle Database Package" on page 34
Register and configure HA for Oracle Database	"How to Register and Configure HA for Oracle Database With Oracle Grid Infrastructure for a Cluster (CLI)" on page 49
Verify the HA for Oracle Database installation	"Verifying the HA for Oracle Database Installation" on page 58
Tune the HA for Oracle Database fault monitor	"Tuning the HA for Oracle Database Fault Monitors" on page 60
(Optional) Customize the HA for Oracle Database server fault monitor	"Customizing the HA for Oracle Database Server Fault Monitor" on page 65
(Optional) Upgrade HA for Oracle Database resource types	"Upgrading HA for Oracle Database Resource Types" on page 75
(Optional) Change the role of an Oracle Data Guard instance	"Changing the Role of an Oracle Data Guard Instance" on page 79

Planning the HA for Oracle Database Installation and Configuration

This section contains the information that you need to plan your HA for Oracle Database installation and configuration.

For information about supported versions, see `Oracle Solaris Cluster 4 Compatibility Guide (http://www.oracle.com/technetwork/server-storage/solaris-cluster/overview/solariscluster4-compatibilityguide-1429037.pdf)` (login to My Oracle Support required).

Configuration Requirements

Caution - Your data service configuration might not be supported if you do not adhere to these requirements.

Use the requirements in this section to plan the installation and configuration of HA for Oracle Database software. These requirements apply to HA for Oracle Database only.

You can configure HA for Oracle Database in a global cluster or a zone cluster.

For requirements that apply to all data services, see "Configuration Guidelines for Oracle Solaris Cluster Data Services" in "Oracle Solaris Cluster Data Services Planning and Administration Guide ".

- **Oracle Grid Infrastructure software requirements -** If you will use Oracle Grid Infrastructure (Oracle ASM and Oracle Clusterware), ensure that the cluster meets Oracle Grid Infrastructure software requirements. See information about Oracle Grid Infrastructure software requirements in the Oracle Grid Infrastructure installation guide for your version of Oracle Grid Infrastructure software.
- **Oracle application files –** These files include Oracle Database binaries, configuration files, and parameter files. You can install these files either on the local file system, the highly available local file system, or on the cluster file system.

  See "Configuration Guidelines for Oracle Solaris Cluster Data Services" in "Oracle Solaris Cluster Data Services Planning and Administration Guide " for the advantages and disadvantages of placing the Oracle Database binaries on the local file system, highly available local file system, and the cluster file system.
- **Database-related files–** These files include the control file, redo logs, and data files. You must install these files on either raw devices or as regular files on the highly available local or cluster file system. For additional information about file systems in a zone cluster, see "Adding File Systems to a Zone Cluster" in "Oracle Solaris Cluster Software Installation Guide ".
- **Oracle ASM configuration –** Oracle Automatic Storage Management (Oracle ASM) is a storage option that provides the services of a file system, logical volume manager, and software redundant array of independent disks (RAID) in a platform independent manner. For more information on Oracle ASM, see the Oracle Database documentation corresponding to the Oracle Database version you are using.

  If the single instance Oracle Database software is installed on an Oracle Solaris Cluster node using an Oracle ASM disk group, then certain files are not included within the Oracle ASM disk group and they reside only locally. Ensure that the following files are copied to the other cluster nodes from the node where the Oracle Database single instance is created:

  - `${ORACLE_BASE}/diag/rdbms/`*sid*
  - `${ORACLE_BASE}/admin`
  - `${ORACLE_HOME}/dbs/pfile.ora`

After you copy these directories to the cluster nodes, ensure that the copied directories have the same user and group permissions as on the source system.

Configuration Planning Questions

Use the questions in this section to plan the installation and configuration of HA for Oracle Database.

- What resource groups will you use for network addresses and application resources and the dependencies between them?
- What is the logical hostname for clients that will access the data service?
- Where will the system configuration files reside?

  See "Configuration Guidelines for Oracle Solaris Cluster Data Services" in "Oracle Solaris Cluster Data Services Planning and Administration Guide " for the advantages and disadvantages of placing the Oracle Database binaries on the local file system rather than the cluster file system.
- Does your database setup require standby instances?

  If you use the `clsetup` utility to register and configure HA for Oracle Database, some of these questions are answered automatically by the utility.

  For information about standby databases, see your Oracle Database documentation.
- Are you planning to use Oracle ASM storage?

  For information about standby databases and Oracle ASM storage, see your Oracle Database documentation.

Preparing the Oracle Solaris Cluster Nodes and Disks

This section contains the procedures that you need to prepare the Oracle Solaris Cluster nodes and disks.

- "How to Prepare the Oracle Solaris Cluster Nodes" on page 19
- "How to Configure Oracle Database Access Using Solaris Volume Manager" on page 21
- "How to Configure Oracle Database Access Using Oracle ASM" on page 22
- "How to Configure an Oracle Grid Infrastructure SCAN Listener" on page 23

▼ How to Prepare the Oracle Solaris Cluster Nodes

Use this procedure to prepare for the installation and configuration of Oracle Database software.

Caution - Perform all of the steps in this section on all the Oracle Solaris Cluster nodes. If you do not perform all of the steps on all of the cluster nodes, the Oracle Database installation is incomplete. An incomplete Oracle Database installation causes HA for Oracle Database to fail during startup.

Note - Consult the Oracle Database documentation before you perform this procedure.

The following steps prepare your cluster nodes and install the Oracle Database software.

1. **Become superuser on all of the cluster members.**

2. **Configure the cluster file system for HA for Oracle Database.**

   - If a cluster file system that is configured using raw devices contains the databases, configure the global devices for raw device access. See the "Oracle Solaris Cluster Software Installation Guide " for information about how to configure global devices.
   - If you use the Solaris Volume Manager software, configure the Oracle Database software to use UNIX file system (UFS) logging on mirrored metadevices or raw-mirrored metadevices. See the Solaris Volume Manager documentation for more information about how to configure raw-mirrored metadevices.
   - If you use an Oracle Solaris ZFS file system for Oracle Database files, configure a highly available local ZFS file system. For more information, see "How to Set Up the HAStoragePlus Resource Type to Make a Local Solaris ZFS File System Highly Available" in "Oracle Solaris Cluster Data Services Planning and Administration Guide ".
   - If you use NFS shares from an Oracle ZFS Storage Appliance NAS device, configure the NAS device with fencing support and the appropriate mount options. See "Requirements When Configuring Oracle ZFS Storage Appliance NAS Devices for Oracle RAC or HA Oracle" in "Oracle Solaris Cluster With Network-Attached Storage Device Manual ".

   For additional information about file systems in a zone cluster, see "Adding File Systems to a Zone Cluster" in "Oracle Solaris Cluster Software Installation Guide ".

3. **Prepare the `$ORACLE_HOME` directory on a local or multihost disk.**

   Note - If you install the Oracle Database binaries on a local disk, use a separate disk than operating system uses, if possible. Installing the Oracle Database binaries on a separate disk prevents the binaries from overwrites during operating environment reinstallation.

4. **Configure kernel parameters in the global cluster and, if used, the zone cluster that runs HA for Oracle Database.**

   You might need to reboot the cluster to initiate certain parameter changes. For information about tuning Oracle Solaris kernel parameters, see "Tuning the Oracle Solaris Kernel" in "Oracle Solaris 11.2 Tunable Parameters Reference Manual ".

For information about shared memory requirements, see the Oracle Database installation guide corresponding to the Oracle Database version you are using for information about configuring kernel parameters in Oracle Solaris software.

5. **If you are using a zone cluster, configure the `limitpriv` property by using the `clzonecluster` command.**

   The `limitpriv` property is required.

   ```
   # clzonecluster configure zcname
   clzonecluster:zcname>set limitpriv="default,proc_priocntl,proc_clock_highres"
   clzonecluster:zcname>commit
   ```

6. **For each zone-cluster node, prevent Oracle Clusterware time synchronization from running in active mode.**

   a. **In the global zone, ensure that the `config/slew_always` property of the NTP service is set to `true`.**

      ```
      # svccfg -s svc:/network/ntp:default listprop config/slew_always
      config/slew_always boolean     true
      ```

      If the property is not set to `true`, use the following commands to set it.

      ```
      # svccfg -s svc:/network/ntp:default setprop config/slew_always = true
      # svcadm refresh svc:/network/ntp:default
      ```

   b. **Log in to the zone-cluster node as root.**

   c. **Create an empty `/etc/inet/ntp.conf` file.**

      ```
      # touch /etc/inet/ntp.conf
      ```

▼ How to Configure Oracle Database Access Using Solaris Volume Manager

Use this procedure to configure Oracle Database using Solaris Volume Manager.

Note - You can run this procedure only in the global zone.

---

1. **Configure the disk devices for the Solaris Volume Manager software to use.**

   See “Oracle Solaris Cluster Software Installation Guide ” for information about how to configure the Solaris Volume Manager software.

2. **If you use raw devices to contain the databases, run the following commands to change each raw-mirrored metadevice's owner, group, and mode.**
   If you do not use raw devices, do not perform this step.

   a. **If you create raw devices, run the following commands for each device on each cluster node that can master the Oracle Database resource group.**

   ```
   # chown oracle /dev/md/metaset/rdsk/dn
   # chgrp dba /dev/md/metaset/rdsk/dn
   # chmod 600 /dev/md/metaset/rdsk/dn
   ```

   metaset

   Specifies the name of the disk set

   `/rdsk/dn`

   Specifies the name of the raw disk device within the *metaset* disk set

   b. **Verify that the changes are effective.**

   ```
   # ls -lL /dev/md/metaset/rdsk/dn
   ```

How to Configure Oracle Database Access Using Oracle ASM

Use this procedure to configure Oracle Database access using Oracle ASM. You can use Oracle ASM on Solaris Volume Manager.

Note - If you use Oracle ASM in a global cluster as well as in a zone cluster configured in that global cluster, you must ensure that, in each particular cluster, Oracle ASM can see only those devices that are intended for its use, whether in the global zone or in a zone cluster. If Oracle ASM can see devices that are used by Oracle ASM in a different cluster, this can cause start problems for Oracle Clusterware or Oracle Grid Infrastructure, because Oracle ASM sees the devices as already mounted elsewhere.

---

1. **Configure a Support for Oracle RAC framework resource group.**
   The Oracle RAC framework resource group is used by the HA for Oracle Database data service to enable Oracle Solaris Cluster and Oracle Clusterware software to interact with each other.
   Follow procedures in “Registering and Configuring the Support for Oracle RAC Framework Resource Group” in “Oracle Solaris Cluster Data Service for Oracle Real Application Clusters Guide ”.

2. **Configure the disk devices for Oracle ASM software to use.**

See "Using Oracle ASM" in "Oracle Solaris Cluster Data Service for Oracle Real Application Clusters Guide " for information about how to configure Oracle ASM.

3. **Set up the permissions for the Oracle ASM disks to be used by Oracle ASM disk groups.**

   a. **Run the following commands for each DID device on each Oracle Solaris Cluster node that can master the Oracle Database resource group.**

   ```
   # chown oraasm:dba /dev/did/rdisk/dn
   # chmod 660 /dev/did/rdisk/dn
   ```

   b. **Verify that the changes are effective.**

   ```
   # ls -lhL /dev/did/rdisk/dn
   ```

How to Configure an Oracle Grid Infrastructure SCAN Listener

When the HA for Oracle Database data service is deployed using Oracle Grid Infrastructure for Clusters, the preferred listener for a single-instance database is to use the Single Client Access Name (SCAN) listener. To leverage a SCAN listener in the HA for Oracle Database configuration, you must ensure that the single-instance database `local_listener` and `remote_listener` parameters are correctly set.

If you configure a SCAN listener, you can ignore subsequent instructions in this manual about creating an Oracle Database listener resource with the `SUNW.oracle_listener` resource type. An Oracle Database listener resource is unnecessary when the SCAN listener is configured.

For more information about the SCAN listener, see the Oracle Grid Infrastructure installation guide for your version of Oracle Grid Infrastructure software.

1. **As the database user, log in to Oracle Database using SQL*Plus.**

2. **Ensure that the `local_listener` parameter contains the local listener IP number and port number.**

   a. **Display the `local_listener` parameter settings.**

   ```
   SQL> show parameters local_listener
   NAME                                 TYPE        VALUE
   ------------------------------------ ----------------------------------------
   local_listener                       string      (ADDRESS=(PROTOCOL=TCP)(HOST=node-IP-
   number)(PORT=port-number))
   ```

```
SQL>
```

b. If the `local_listener` parameter does not contain the local listener IP number and port number, set the parameter to a blank value and restart the Oracle database.

When the Oracle database is restarted, Oracle Clusterware dynamically sets the `local_listener` parameter.

```
SQL> alter system set local_listener='' scope=both;
System altered.
```

c. Verify that the `local_listener` parameter settings are correct.

```
SQL> show parameters local_listener
```

3. Ensure that the `remote_listener` parameter refers to the SCAN name and port number.

a. Determine whether the `remote_listener` parameter contains the SCAN name and port number.

```
phys-schost$ srvctl config scan
SCAN name: SCAN-name, Network: network-number
…
phys-schost$ srvctl config scan_listener
SCAN Listener LISTENER_SCAN1 exists. Port: TCP:port-number
…
```

b. Determine whether the `remote_listener` parameter contains the SCAN name and port number.

```
SQL> show parameter remote_listener

NAME                                 TYPE        VALUE
------------------------------------ ----------- ------------------------------
remote_listener                      string       SCAN-name:port-number
SQL>
```

c. If the `remote_listener` parameter does not contain the SCAN name and port number, set the parameter with those values.

```
SQL> alter system set remote_listener='SCAN-name:port-number' scope=both
```

d. Verify the SCAN settings in the `remote_listener` parameter.

```
SQL> show parameter remote_listener
```

4. On each cluster node, ensure that the `tnsnames.ora` file refers to the SCAN name for the Net Service Name.

```
ORCL =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = SCAN-name)(PORT = port-number))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = orcl)
    )
  )
```

Using a VUCMM Framework Resource Group

HA for Oracle Database can be configured to use a Sun QFS file system.

The data service uses a resource group that is based on the Oracle Solaris Cluster volume manager reconfiguration framework (VUCMM) resource type, `SUNW.vucmm_framework`. The `SUNW.vucmm_framework` resource type is a single-instance resource type. You can create only one resource of this type in the cluster.

▼ How to Use a VUCMM Framework Resource Group

1. **Become superuser on any cluster node.**

2. **Create a scalable VUCMM framework resource group.**

   ```
   # clresourcegroup create -n nodelist-S vucmm-fmwk-rg
   ```

   -n `nodelist`=*nodelist*

   Specifies a comma-separated list of cluster nodes on which HA for Oracle Database is to be enabled. The HA for Oracle Database software packages must be installed on each node in this list.

   vucmm-fmwk-rg

   Specifies the name that you are assigning to the resource group.

3. **Register the `SUNW.vucmm_framework` resource type.**

   ```
   # clresourcetype register SUNW.vucmm_framework
   ```

4. **Add an instance of the `SUNW.vucmm_framework` resource type to the resource group that you created in Step 2.**

   ```
   # clresource create -g vucmm-fmwk-rg -t SUNW.vucmm_framework vucmm-fmwk-rs
   ```

vucmm-fmwk-rs

Specifies the name that you are assigning to the `SUNW.vucmm_framework` resource.

5. **Register and add an instance of the resource type that represents the volume manager that you are using for Oracle Database files, if any.**

   - **If you are using Solaris Volume Manager for Solaris Cluster, register and add the instance as follows:**

     a. **Register the `SUNW.vucmm_svm` resource type.**

     ```
     # clresourcetype register SUNW.vucmm_svm
     ```

     b. **Add an instance of the `SUNW.vucmm_svm` resource type to the resource group that you created in Step 2.**

     Ensure that this instance depends on the `vucmm_framework` resource that you created in Step 4.

     ```
     # clresource create -g vucmm-fmwk-rg \
     -t SUNW.vucmm_svm \
     -p resource_dependencies=vucmm-fmwk-rs  vucmm-svm-rs
     ```

     `-p resource_dependencies=`*vucmm-fmwk-rs*

     Specifies that this instance depends on the `SUNW.vucmm_framework` resource.

     vucmm-svm-rs

     Specifies the name that you are assigning to the `SUNW.vucmm_svm` resource.

6. **Verify the configuration of the VUCMM framework resource group.**

   ```
   # clresourcegroup show vucmm-fmwk-rg
   ```

7. **Bring online and in a managed state the VUCMM framework resource group and its resources.**

   ```
   # clresourcegroup online -emM vucmm-fmwk-rg
   ```

   vucmm-fmwk-rg

   Specifies the name of the `SUNW.vucmm_framework` based resource group.

8. **Verify that all resource groups are online.**

   ```
   # clresourcegroup status
   ```

Installing Oracle Grid Infrastructure Software

Oracle Grid Infrastructure installation consists of installing Oracle Grid Infrastructure, which installs Oracle ASM on cluster nodes, configuring the Oracle ASM disk groups, and starting Oracle ASM instances. An Oracle ASM disk group is a collection of disk devices to store data files that Oracle ASM instances manage as a unit. Oracle ASM instances mount disk groups to make Oracle ASM files available to database instances.

When you run Oracle Universal Installer, you have the option to configure and install Oracle Grid Infrastructure and create Oracle ASM disk groups. For detailed information, see the *Oracle Automatic Storage Management Administrator's Guide* corresponding to the Oracle Grid Infrastructure version you are using.

Note - Before starting Oracle Universal Installer, ensure that the Oracle Solaris `library/motif` package is installed.

For zone clusters, also ensure that the `group/system/solaris-large-server` package is installed.

Note - Set up and configure the file system, where the Oracle Grid Infrastructure `ORACLE_HOME` resides, before performing Oracle Grid Infrastructure installation. Use only a local file system for the Oracle Grid Infrastructure home.

Verifying Oracle Grid Infrastructure Software Installation

After installing Oracle Grid Infrastructure software, you can verify that Oracle Grid Infrastructure software is installed and the Oracle ASM disk group is mounted on a cluster node by performing the following steps on the Oracle ASM instance.

```
# sqlplus  "/ as sysasm"
sql> select * from v$sga;
sql> select name,state from v$asm_diskgroup;
sql> exit;
#
```

Installing Oracle Database Software

This section contains the procedures that you need to install Oracle Database software.

- “How to Install Oracle Database Software” on page 28

- "How to Set the Oracle Database Kernel Parameters" on page 28

How to Install Oracle Database Software

Before You Begin Ensure that Oracle Grid Infrastructure is installed. See "Installing Oracle Grid Infrastructure Software" on page 27.

1. **Become superuser on a cluster member.**

2. **If you plan to install Oracle Database software on a cluster file system, start the Oracle Solaris Cluster software and become the owner of the device group.**

   If you plan to install Oracle Database software at another location, omit this step.

   For more information about installation locations, see "Preparing the Oracle Solaris Cluster Nodes and Disks" on page 19.

3. **Install Oracle Database software.**

   Before you start the Oracle Database installation, ensure that the system resources required for Oracle Database are configured. Log in as `oracle` to ensure ownership of the entire directory before you perform this step. See the appropriate Oracle Database installation and configuration guides for instructions about how to install Oracle Database software.

Tip - You can use Oracle Solaris Resource Management (SRM) to ensure that the kernel parameters are set to at least the minimum values that Oracle Database software requires. For more information about setting the Oracle Database kernel parameters, see "How to Set the Oracle Database Kernel Parameters" on page 28. After the system resources are configured for Oracle Database, you can start the installation itself.

How to Set the Oracle Database Kernel Parameters

The `default` project is modified to contain the resources required for Oracle Database as the RGM uses the `default` project for running the data service. If you want to use a specific SRM project for running Oracle Database, you must create that project and configure the system resources in that project using the same procedure. Specify the project name instead of `default`. When you configure the resource group or resource for the Oracle Database server, specify that project name in the corresponding property of the resource group or resource.

1. **Display the settings for the `default` project.**

   ```
   phys-X# prctl -i project default
   ```

2. **If no kernel parameters are set, or if any kernel parameters are not set to the minimum required value for Oracle Database as shown in the following table, set the parameter.**

```
phys-X# projmod -s -K "parameter=(priv,value,deny)" default
```

Oracle Database Kernel Parameter	Minimum Required Value
process.max-sem-nsems	256
project.max-sem-ids	100
project.max-shm-ids	100
project.max-shm-memory	4294967295

See the installation guide for your version of Oracle Database for more information about these parameters.

3. **Verify the new settings.**

```
phys-X# prctl -i project default
```

4. **Set the `noexec_user_stack` parameter.**

Open the `/etc/system` file and manually add the following entry.

```
set noexec_user_stack=1
```

Verifying Oracle Database Installation

This section contains the procedure to verify Oracle Database installation.

▼ How to Verify Oracle Database Installation

This procedure does not verify that your application is highly available because you have not yet installed your data service.

1. **Confirm that the owner, group, and mode of the `$ORACLE_HOME/bin/oracle` file are as follows:**

- Owner: `oracle`
- Group: `dba`
- Mode: `-rwsr-s--x`

```
# ls -l $ORACLE_HOME/bin/oracle
```

2. **Verify that the listener binaries exist in the `$ORACLE_HOME/bin` directory.**

Next Steps When you have completed the work in this section, go to "Creating an Oracle Database" on page 30.

Creating an Oracle Database

After verifying the Oracle Database installation, create the Oracle databases that you require.

- If you are using Oracle Database *without* standby databases, perform the procedure "How to Create a Primary Oracle Database" on page 30.
- If you are using Oracle Data Guard, create the following database instances:
  - **Primary database instance**. For instructions for creating a primary database, see "How to Create a Primary Oracle Database" on page 30.
  - **Standby database instances**. A standby database instance can be either a physical standby database instance or a logical standby database instance. For instructions for creating standby database instances, see your Oracle Database documentation.

▼ How to Create a Primary Oracle Database

1. **Prepare database configuration files.**
   Place all of the database-related files (data files, redo log files, and control files) on either shared raw global devices or on the cluster file system. See "Preparing the Oracle Solaris Cluster Nodes and Disks" on page 19 for information about installation locations.

   Note - If the database exists in the non-global zone, do not place the database-related files on the shared raw devices.

   Within the `init$ORACLE_SID.ora` or `config$ORACLE_SID.ora` file, you might need to modify the assignments for `control_files` and `background_dump_dest` to specify the locations of the control files and alert files.

   Note - If you use Oracle Solaris authentication for database logins, set the `remote_os_authent` variable in the `init$ORACLE_SID.ora` file to `True`.

2. **Start the creation of the database by using a supported Oracle Database creation utility.**

During creation, ensure that all the database-related files are placed in the appropriate location, either on shared global devices, on the cluster file system, or on a highly available local file system.

3. **Verify that the file names of your control files match the file names in your configuration files.**

4. **Create the `v$sysstat` view.**

   Run the catalog scripts that create the `v$sysstat` view. The HA for Oracle Database fault monitor uses this view. For more information, see your Oracle Database documentation.

Next Steps When you have completed the work in this section, go to "Setting Up Oracle Database Permissions" on page 31.

Setting Up Oracle Database Permissions

Caution - Do not perform the steps in this section for an Oracle Database physical standby database.

Perform the procedure in this section to set up database permissions for an Oracle primary database or an Oracle Database logical standby database.

▼ How to Set Up Oracle Database Permissions

1. **Enable access for the user and password to be used for fault monitoring.**

   - **To use the Oracle Database authentication method, grant to this user authority on the `v_$sysstat` view and the `v_$archive_dest` view.**

```
# sqlplus  "/ as sysdba"

sql> create user user identified by passwd;
sql> alter user user default tablespace system quota 1m on system;
sql> grant select on v_$sysstat to user;
sql> grant select on v_$archive_dest to user;
sql> grant select on v_$database to user;
sql> grant create session to user;
sql> grant create table to user;
sql> create profile profile limit PASSWORD_LIFE_TIME UNLIMITED;
sql> alter user user identified by passwd profile profile;
```

```
sql> exit;
#
```

You can use this method for all supported Oracle Database releases.

- **To use the Oracle Solaris authentication method, perform the following steps:**

  a. **Confirm that the `remote_os_authent` parameter is set to `TRUE`.**

  ```
  # sqlplus  "/ as sysdba"
  sql> show parameter remote_os_authent

  NAME                          TYPE        VALUE
  ----------------------------- ----------- ---------------
  remote_os_authent             boolean     TRUE
  ```

  b. **Determine the setting of the `os_authent_prefix` parameter.**

  ```
  # sql>  show parameter os_authent_prefix

  NAME                          TYPE        VALUE
  ----------------------------- ----------- ---------------
  os_authent_prefix             string      ops$
  ```

  c. **Grant permission for the database to use Oracle Solaris authentication.**

  ```
  sql> create user prefix user identified by externally default
  tablespace system quota 1m on system;
  sql> grant connect, resource to prefix user;
  sql> grant select on v_$sysstat to prefix user;
  sql> grant select on v_$archive_dest to prefix user;
  sql> grant select on v_$database to prefix user;
  sql> grant create session to prefix user;
  sql> grant create table to prefix user;
  sql> exit;
  #
  ```

  The replaceable items in these commands are as follows:

  - *prefix* is the setting of the `os_authent_prefix` parameter. The default setting of this parameter is `ops$`.
  - *user* is the user for whom you are enabling Oracle Solaris authentication. Ensure that this user owns the files under the `$ORACLE_HOME` directory.

Note - Do not type a space between *prefix* and *user*.

2. **Configure Oracle Net for the Oracle Solaris Cluster software.**

The `listener.ora` file must be accessible from all the nodes that are in the cluster. Place these files either under the cluster file system or in the local file system of each cluster node that can potentially run the Oracle Database resources.

Note - If you place the `listener.ora` file in a location other than the `/var/opt/oracle` directory or the `$ORACLE_HOME/network/admin` directory, you must specify the `TNS_ADMIN` variable or an equivalent Oracle Database variable in a user-environment file. For information about Oracle Database variables, see the Oracle Database documentation.

You must also run the `clresource` command to set the resource extension parameter `User_env`, which sources the user-environment file. See "`SUNW.oracle_listener` Extension Properties" on page 85 or "`SUNW.oracle_server` Extension Properties" on page 81 for format details.

HA for Oracle Database imposes no restrictions on the listener name; it can be any valid Oracle Database listener name.

The following code sample identifies the lines in `listener.ora` that are updated.

```
LISTENER =
(DESCRIPTION_LIST =
(DESCRIPTION =
(ADDRESS_LIST =
(ADDRESS =  (PROTOCOL = IPC) (KEY = EXTPROC))
(ADDRESS =  (PROTOCOL = TCP) (HOST = logical-hostname) (PORT = port-used))
)
)
```

The following code sample identifies the lines in `tnsnames.ora` that are updated on client machines.

```
service_name =
.
.
(ADDRESS =
(PROTOCOL = TCP)
(HOST = logicalhostname) <- logical hostname
(PORT = 1527) <- must match port in LISTENER.ORA
)
)
(CONNECT_DATA =
(SID = <SID>)) <- database name, default is ORCL
```

3. Verify that the Oracle Solaris Cluster software is installed and running on all the nodes.

```
# cluster status clustername
```

Next Steps Go to "Installing the HA for Oracle Database Package" on page 34 to install the HA for Oracle Database packages.

Installing the HA for Oracle Database Package

If you did not install the HA for Oracle Database package during your initial Oracle Solaris Cluster installation, perform this procedure to install the package.

Note - You must install the HA for Oracle Database package in the global cluster, even if you plan to run HA for Oracle Database in a zone cluster.

▼ How to Install the HA for Oracle Database Package

Perform this procedure on each cluster node where you want the HA for Oracle Database software to run.

1. **On the cluster node where you are installing the data service package, assume the `root` role.**

2. **Ensure that the data service package is available from the configured publisher and that the `solaris` and `ha-cluster` publishers are valid.**

```
# pkg list -a ha-cluster/data-service/oracle-database
# pkg publisher
PUBLISHER                             TYPE     STATUS   P  LOCATION
solaris                               origin   online   F  solaris-repository
ha-cluster                            origin   online   F  ha-cluster-repository
```

For information about setting the `solaris` publisher, see "Adding, Modifying, or Removing Package Publishers" in "Adding and Updating Software in Oracle Solaris 11.2".

Tip - Use the `-nv` options whenever you install or update to see what changes will be made, such as which versions of which packages will be installed or updated and whether a new BE will be created.

If you do not get any error messages when you use the `-nv` options, run the command again without the `-n` option to actually perform the installation or update. If you do get error messages, run the command again with more `-v` options (for example, `-nvv`) or more of the package FMRI pattern to get more information to help you diagnose and fix the problem. For troubleshooting information, see Appendix A, "Troubleshooting Package Installation and Update," in "Adding and Updating Software in Oracle Solaris 11.2".

3. **Install the HA for Oracle Database software package.**

```
# pkg install ha-cluster/data-service/oracle-database ha-cluster/library/ucmm
```

4. **Verify that the package installed successfully.**

```
$ pkg info ha-cluster/data-service/oracle-database ha-cluster/library/ucmm
```

Installation is successful if output shows that `State` is `Installed`.

5. **Perform any necessary updates to the Oracle Solaris Cluster software.**

For instructions on updating your software, see Chapter 11, “Updating Your Software,” in “Oracle Solaris Cluster System Administration Guide”.

Registering and Configuring HA for Oracle Database

This section describes how to register and configure the HA for Oracle Database data service, with or without Oracle Grid Infrastructure software installed on the same nodes that run Oracle Solaris Cluster software.

This section provides the following information:

Tools for Registering and Configuring HA for Oracle Database

Oracle Solaris Cluster provides the following tools for registering and configuring HA for Oracle Database:

- **The `clsetup` utility.** For more information, see “How to Register and Configure HA for Oracle Database (`clsetup`)” on page 37.
- **Oracle Solaris Cluster Manager.** For more information, see Chapter 13, “Using the Oracle Solaris Cluster GUI,” in “Oracle Solaris Cluster System Administration Guide ”.

- **Oracle Solaris Cluster maintenance commands.** For more information, see "How to Register and Configure HA for Oracle Database Without Oracle Grid Infrastructure (CLI)" on page 41.

The `clsetup` utility *and the Oracle Solaris Cluster Manager graphical user interface (GUI) each provide* a wizard for configuring HA for Oracle Database. The wizard reduces the possibility for configuration errors that might result from command syntax errors or omissions. This wizard also ensures that all required resources are created and that all required dependencies between resources are set.

Setting HA for Oracle Database Extension Properties

Use the extension properties in Appendix A, "HA for Oracle Database Extension Properties" to create your resources. To set an extension property of a resource, include the option `-p` *property=value* in the `clresource` command that creates or modifies the resource. Use the procedure in Chapter 2, "Administering Data Service Resources," in "Oracle Solaris Cluster Data Services Planning and Administration Guide " to configure the extension properties if you have already created your resources.

You can update some extension properties dynamically. You can update others, however, only when you create or disable a resource. The Tunable entries indicate when you can update each property. See the `r_properties`(5) man page for details about all Oracle Solaris Cluster resource properties.

"`SUNW.oracle_server` Extension Properties" on page 81 describes all extension properties that you can set for the Oracle Database server. The following extension properties are required for the Oracle Database server.

- If using Oracle Grid Infrastructure:
  - `Db_unique_name`
  - `ORACLE_HOME`
  - `ORACLE_SID`
- If not using Oracle Grid Infrastructure:
  - `Alert_log_file`
  - `Connect_string`
  - `Db_unique_name`
  - `ORACLE_HOME`
  - `ORACLE_SID`

▼ How to Register and Configure HA for Oracle Database (clsetup)

This procedure uses the clsetup configuration wizard for HA for Oracle Database.

Note - The clsetup utility does not support configuration of HA for Oracle Database with ZFS.

Before You Begin Ensure that the following prerequisites are met:

- The volume manager of the cluster is configured to provide volumes on shared storage that are accessible from any Oracle Solaris Cluster node where Oracle Database could potentially run.
- Raw devices and file systems on the storage volumes that Oracle Database will use for its database are created.
- Oracle Database software is installed to be accessible from all nodes where Oracle Database could potentially run.
- Kernel variables for the UNIX operating system are configured for Oracle Database.
- Oracle Database software is configured for all nodes that could potentially run Oracle Database.
- The data service packages are installed.

Ensure that you have the following information:

- The names of the cluster nodes that master the data service.
- The path to the Oracle Database application binaries for the resources that you plan to configure.
- The database type.

1. **Become superuser on any cluster node.**

2. **Start the clsetup utility.**

```
# clsetup
```

The clsetup main menu is displayed.

3. **Type the number that corresponds to the option for data services and press Return.**

The Data Services menu is displayed.

4. **Type the number that corresponds to the option for configuring HA for Oracle Database and press Return.**

The clsetup utility displays the list of prerequisites for performing this task.

5. **Verify that the prerequisites are met, and press Return.**

   The clsetup utility displays a list of the cluster nodes.

6. **Select the nodes where you require Oracle Database to run.**

   - **To accept the default selection of all listed nodes in an arbitrary order, press Return.**
   - **To select a subset of the listed nodes, type a comma-separated or space-separated list of the numbers that correspond to the nodes. Then press Return.**

     Ensure that the cluster nodes are listed in the order in which the nodes are to appear in the node list of the resource group in which the Oracle Database resource is to be placed.
   - **To select all cluster nodes in a particular order, type a comma-separated or space-separated ordered list of the numbers that correspond to the nodes and press Return.**

     Ensure that the cluster nodes are listed in the order in which the nodes are to appear in the node list of the resource group in which the Oracle Database resource is to be placed.

7. **To confirm your selection of cluster nodes, type d and press Return.**

   The clsetup utility displays the types of Oracle Database components that are to be configured.

8. **Type the numbers of the Oracle Database components you want to configure and press Return.**

   The clsetup utility lists the Oracle Database home directory.

9. **Specify the Oracle Database home directory for your installation of the Oracle Database software.**

   - **If the directory is listed, select the directory as follows:**

     a. **Type the number that corresponds the directory that you are selecting.**

        The clsetup utility displays a list of Oracle Database system identifiers that are configured on the cluster. The utility also prompts you to specify the system identifier for your installation of Oracle Database.
   - **If the directory is not listed, specify the directory explicitly.**

     a. **Type e and press Return.**

        The clsetup utility prompts you for the Oracle Database home directory.

b. Type the full path to the Oracle Database home directory and press Return.

The clsetup utility displays a list of Oracle Database system identifiers that are configured on the cluster. The utility also prompts you to specify the system identifier for your installation of Oracle Database.

10. Specify the Oracle Database SID of the Oracle database that you are configuring.

- **If the SID is listed, select the SID as follows:**

  a. Type the number that corresponds the SID that you are selecting.

  The clsetup utility displays the properties of the Oracle Solaris Cluster resources that the utility will create.

- **If the SID is not listed, specify the SID explicitly.**

  a. Type e and press Return.

  The clsetup utility prompts you for the SID.

  b. Type the SID and press Return.

  The clsetup utility displays the properties of the Oracle Solaris Cluster resources that the utility will create.

The clsetup utility displays the properties of the Oracle Solaris Cluster resources that the utility will create.

11. If you require a different name for any Oracle Solaris Cluster resources properties, change each value as follows.

a. Type the number that corresponds to the name that you are changing and press Return.

The clsetup utility displays a screen where you can specify the new name.

b. At the New Value prompt, type the new name and press Return.

The clsetup utility returns you to the list of the properties of the Oracle Solaris Cluster resource that the utility will create.

12. To confirm your selection of Oracle Solaris Cluster resource properties, type d and press Return.

The clsetup utility displays a list of existing storage resources. If no storage resources are available, the clsetup utility displays a list of shared storage types where data is to be stored.

13. **Type the numbers that correspond to type of shared storage that you are using for storing the data and press Return.**

    The clsetup utility displays the file-system mount points that are configured in the cluster.

14. **Select the file system mount points as follows.**

    - **To accept the default selection of all listed file-system mount points in an arbitrary order, type a. Then press Return.**
    - **To select a subset of the listed file system mount points, type a comma-separated or space-separated list of the numbers that correspond to the file-system mount points. Then press Return.**

    The clsetup utility displays the global disk sets and device groups that are configured in the cluster.

15. **Select the device groups as follows.**

    - **To accept the default selection of all listed device groups in an arbitrary order, type a and press Return.**
    - **To select a subset of the listed device groups, type a comma-separated or space-separated list of the numbers that correspond to the device groups and press Return.**

    The clsetup utility returns to you the list of highly available storage resources.

16. **Type a comma-separated or space-separated list of the numbers that correspond to the storage resources that your data service requires, and press Return.**

17. **To confirm your selection of Oracle Solaris Cluster storage resources, type d and press Return.**

    The clsetup utility displays all the existing logical hostname resources in the cluster. If there are no logical hostname resources available, the clsetup utility prompts for the logical hostname that the resource will make highly available.

18. **Specify the logical hostname and press Return.**

    The clsetup utility returns to you the list of available logical hostname resources.

19. **Type a comma-separated or space-separated list of the numbers that correspond to the logical hostname resources that your data service requires, and press Return.**

20. **To confirm your selection of Oracle Solaris Cluster logical hostname resources, type d and press Return.**

The `clsetup` utility displays the names of the Oracle Solaris Cluster objects that the utility will create.

21. **If you require a different name for any Oracle Solaris Cluster objects, change each name as follows.**

    - **Type the number that corresponds to the name that you are changing and press Return.**
      The `clsetup` utility displays a screen where you can specify the new name.
    - **At the New Value prompt, type the new name and press Return.**
      The `clsetup` utility returns you to the list of the names of the Oracle Solaris Cluster objects that the utility will create.

22. **To confirm your selection of Oracle Solaris Cluster object names, type `d` and press Return.**

23. **To create the configuration, type `c` and Press Return.**
    The `clsetup` utility displays a progress message to indicate that the utility is running commands to create the configuration. When configuration is complete, the `clsetup` utility displays the commands that the utility ran to create the configuration.

24. **Press Return to continue.**

25. **(Optional) Type `q` and press Return repeatedly until you quit the `clsetup` utility.**
    If you prefer, you can leave the `clsetup` utility running while you perform other required tasks before using the utility again.

▼ How to Register and Configure HA for Oracle Database Without Oracle Grid Infrastructure (CLI)

This procedure provides the long forms of the Oracle Solaris Cluster maintenance commands. Most commands also have short forms. Except for the forms of the command names, the commands are identical.

Before You Begin Ensure that the following prerequisites are met:

- The `/etc/netmasks` file has IP-address subnet and netmask entries for all logical hostnames. If necessary, edit the `/etc/netmasks` file to add any missing entries.
- If you are using a volume manager, the volume manager of the cluster is configured to provide volumes on shared storage that are accessible from any Oracle Solaris Cluster node where Oracle Database software could potentially run.

- If you are using a volume manager, raw devices and file systems on the storage volumes that Oracle Database software will use for its database are created.
- Oracle Database software is installed to be accessible from all cluster nodes where Oracle Database could potentially run.
- Kernel variables for the UNIX operating system are configured for Oracle Database.
- Oracle Database software is configured for all cluster nodes that could potentially run Oracle Database.
- The data service packages are installed.

Ensure that you have the following information:

- The names of the cluster nodes that master the data service.
- The logical hostname that clients use to access the data service. Normally, you set up this IP address when you install the cluster. See the "Oracle Solaris Cluster Concepts Guide " for details about network resources.
- The path to the Oracle Database application binaries for the resources that you plan to configure.
- The database type.

1. **On a cluster member, become superuser or assume a role that provides `solaris.cluster.modify` and `solaris.cluster.admin` RBAC authorizations.**

2. **Register the resource types for the data service.**

   For HA for Oracle Database, you register two resource types, `SUNW.oracle_server` and `SUNW.oracle_listener`, as follows.

   Note - If you are using an Oracle Grid Infrastructure for Clusters Single Client Access Name (SCAN) listener, omit registration of the `SUNW.oracle_listener` resource type.

   ```
   # clresourcetype register SUNW.oracle_server
   # clresourcetype register SUNW.oracle_listener
   ```

3. **Create a failover resource group to hold the network and application resources.**

   This step is not required if you use the Oracle Solaris ZFS file system, because the resource group was created when the highly available local ZFS file system was configured in "How to Prepare the Oracle Solaris Cluster Nodes" on page 19. The resources that are created in other steps in this procedure are to be added to this resource group.

   You can optionally select the set of cluster nodes on which the data service can run with the `-n` option, as follows.

   ```
   # clresourcegroup create [-n node-zone-list] resource-group
   ```

resource-group

Specifies the name of the resource group. This name can be your choice but must be unique for resource groups within the cluster.

4. **Verify that all of the network resources that you use have been added to your name service database.**

You should have performed this verification during the Oracle Solaris Cluster installation.

Note - Ensure that all of the network resources are present in the server's and client's `/etc/inet/hosts` file to avoid any failures because of name service lookup.

---

5. **Add a logical hostname resource to the failover resource group.**

```
# clreslogicalhostname create -g resource-group [-h logical-hostname] logical-hostname-rs
```

logical-hostname

Specifies a logical hostname. This logical hostname must be present in your name service database. If *logical-hostname* and *logical-hostname-rs* are identical, *logical-hostname* is optional.

logical-hostname-rs

Specifies the name that you are assigning to the logical hostname resource that you are creating.

6. **Register the `SUNW.HAStoragePlus` resource type with the cluster.**

```
# clresourcetype register SUNW.HAStoragePlus
```

7. **Add a resource of type `SUNW.HAStoragePlus` to the failover resource group.**

Note - If you use the Oracle Solaris ZFS file system for Oracle files, omit this step. The `HAStoragePlus` resource was created when the highly available local ZFS file system was configured. For more information, see "How to Prepare the Oracle Solaris Cluster Nodes" on page 19.

Caution - Raw devices from Oracle Solaris Cluster device groups are not supported in non-global zones.

```
# clresource create -g resource-group -t SUNW.HAStoragePlus \
-p GlobalDevicePaths=device-path \
-p FilesystemMountPoints=mount-point-list \
-p AffinityOn=TRUE hastp-rs
```

You must set either the `GlobalDevicePaths` extension property or the `FilesystemMountPoints` extension property:

- If your database is on a raw device, set the `GlobalDevicePaths` extension property to the global device path.
- If your database is on the cluster file system, specify mount points of the cluster file system and the local file system.

Note - `AffinityOn` must be set to `TRUE` and the local file system must reside on global disk groups to be failover.

The resource is created in the enabled state.

8. **Bring online the failover resource group in a managed state on a cluster node.**

```
# clresourcegroup online -M resource-group
```

`-M`

Places the resource group that is brought online in a managed state.

9. **Create Oracle Database application resources in the failover resource group.**

- Oracle Database server resource:

```
# clresource create -g resourcegroup \
-t SUNW.oracle_server \
-p Alert_log_file=path-to-log \
-p Connect_string=user/passwd \
-p Oracle_sid=instance \
-p Oracle_home=Oracle_home \
-p Restart_type=entity-to-restart \
[-p Dataguard_role=role] \
[-p Standby_mode=mode] \
-p Resource_dependencies_offline_restart=storageplus-resource \
resource
```

- Oracle Database listener resource:

```
# clresource create -g resource-group \
-t SUNW.oracle_listener \
-p Listener_name=listener \
-p Oracle_home=Oracle_home \
-p Resource_dependencies_offline_restart=storageplus-resource \
resource
```

`-g` *resource-group*

Specifies the name of the resource group into which the resources are to be placed.

`-t`

Specifies the type of the resource to add.

`-p Alert_log_file =`*path-to-log*

Sets the path under `$ORACLE_HOME` for the server message log.

`-p Connect_string =`*user/passwd*

Specifies the user and password that the fault monitor uses to connect to the database. These settings must agree with the permissions that you set up in "How to Set Up Oracle Database Permissions" on page 31. If you use Oracle Solaris authorization, type a slash (`/`) instead of the user name and password.

`-p Oracle_sid=`*instance*

Sets the Oracle Database system identifier.

`-p Oracle_home=`*Oracle_home*

Sets the path to the Oracle Database home directory.

`-p Listener_name=`*listener*

Sets the name of the Oracle Database listener instance. This name must match the corresponding entry in `listener.ora`.

`-p Restart_type=`*entity-to-restart*

Specifies the entity that the server fault monitor restarts when the response to a fault is restart. Set *entity-to-restart* as follows:

- To specify that only this resource is restarted, set *entity-to-restart* to `RESOURCE_RESTART`. By default, only this resource is restarted.
- To specify that all resources in the resource group that contains this resource are restarted, set *entity-to-restart* to `RESOURCE_GROUP_RESTART`.

  If you set *entity-to-restart* to `RESOURCE_GROUP_RESTART`, all other resources (such as Apache or DNS) in the resource group are restarted, even if they are not faulty. Therefore, include in the resource group only the resources that you require to be restarted when the Oracle Database server resource is restarted.

`-p Dataguard_role=`*role*

Specifies the role of the database instance. Change *role* as follows:

- To create a resource for a primary database instance that does not have standby instances configured, change *role* to `NONE`. This value is the default value.

- To create a resource for a primary database instance that has standby database instances configured, change *role* to `PRIMARY`.
- To create a resource for a standby database instance, change *role* to `STANDBY`.

`-p Standby_mode=`*mode*

Specifies the mode for the standby database instance. If you change `Dataguard_role` to `NONE` or `PRIMARY`, the value of the `Standby_mode` is ignored.

- To specify a logical standby database, change *mode* to `LOGICAL`. This value is the default value.
- To specify a physical standby database, change *mode* to `PHYSICAL`.
- To specify a snapshot standby database, change *mode* to `SNAPSHOT`.

resource

Specifies the name of the resource that you are creating.

Note - Optionally, you can set additional extension properties that belong to the Oracle Database data service to override their default values. See "Setting HA for Oracle Database Extension Properties" on page 36 for a list of extension properties.

The resources are created in the enabled state.

Example 1-1 Registering HA for Oracle Database to Run in the Global Zone

This example shows how to register HA for Oracle Database on a two-node cluster. The following are the sample names used in the commands:

Node names

`phys-schost-1, phys-schost-2`

Logical hostname

`schost-1`

Resource group

`resource-group-1` (failover resource group)

HAStoragePlus resource

`hastp-rs`

Oracle resources

`oracle-server-1, oracle-listener-1`

Oracle instances

`ora-lsnr` (listener), `ora-srvr` (server)

Create the failover resource group to contain all of the resources.

```
# clresourcegroup create resource-group-1
```

Add the logical hostname resource to the resource group.

```
# clreslogicalhostname create -g resource-group-1 schost-1
```

Register the SUNW.HAStoragePlus resource type.

```
# clresourcetype register SUNW.HAStoragePlus
```

Add a resource of type SUNW.HAStoragePlus to the resource group.

```
# clresource create -g resource-group-1 \
-t SUNW.HAStoragePlus \
-p FileSystemMountPoints=/global/oracle,/global/ora-data/logs,/local/ora-data \
-p AffinityOn=TRUE \
hastp-rs
```

Bring the resource group online in a managed state

```
# clresourcegroup online -M resource-group-1
```

Register the Oracle Database resource types.

```
# clresourcetype register SUNW.oracle_server
# clresourcetype register SUNW.oracle_listener
```

Add the Oracle Database application resources to the resource group.

```
# clresource create -g resource-group-1 \
-t SUNW.oracle_server \
-p Alert_log_file=/global/oracle/message-log \
-p Connect_string=scott/tiger \
-p Oracle_home=/global/oracle \
-p Oracle_sid=ora-srvr \-p Dataguard_role=STANDBY \
-p Standby_mode=PHYSICAL \
-p Resource_dependencies_offline_restart=hastp-rs \
oracle-server-1

# clresource create -g resource-group-1 \
-t SUNW.oracle_listener \
-p Oracle_home=/global/oracle \
-p Listener_name=ora-lsnr \
oracle-listener-1
```

Example 1-2 Registering HA for Oracle Database to Run in a Zone Cluster

This example shows how to register HA for Oracle Database in a zone cluster. The following are the sample names used in the commands, which are issued from the global cluster:

Node names

```
phys-schost-1, phys-schost-2
```

Zone cluster names

```
zonecluster1, zonecluster2
```

Logical hostname

`zchost-1`

Resource group

`resource-group-1` (failover resource group)

HAStoragePlus resource

`hastp-rs`

Oracle resources

`oracle-server-1, oracle-listener-1`

Oracle instances

`ora-lsnr` (listener), `ora-srvr` (server)

Create the failover resource group to contain all of the resources.)

```
# clresourcegroup create -Z zonecluster1 resource-group-1
```

Add the logical hostname resource to the resource group.)

```
# clreslogicalhostname create -Z zonecluster1 -g resource-group-1 zchost-1
```

Register the SUNW.HAStoragePlus resource type.)

```
# clresourcetype register -Z zonecluster1 SUNW.HAStoragePlus
```

Add a resource of type SUNW.HAStoragePlus to the resource group.)

```
# clresource create -Z zonecluster1 \
-g resource-group-1 \
-t SUNW.HAStoragePlus \
-p FileSystemMountPoints=/global/oracle,/global/ora-data/logs,/local/ora-data \
-p AffinityOn=TRUE \
hastp-rs
```

Bring the resource group online in a managed state

```
# clresourcegroup online -Z zonecluster1 -M resource-group-1
```

Register the Oracle Database resource types.)

```
# clresourcetype register -Z zonecluster1 SUNW.oracle_server
# clresourcetype register -Z zonecluster1 SUNW.oracle_listener
```

Add the Oracle Database application resources to the resource group.)

```
# clresource create -Z zonecluster1 \
-g resource-group-1 \
-t SUNW.oracle_server \
-p Alert_log_file=/global/oracle/message-log \
-p Connect_string=scott/tiger \
-p Oracle_home=/global/oracle \
-p Oracle_sid=ora-srvr \
-p Dataguard_role=STANDBY \
-p Standby_mode=PHYSICAL \
```

```
oracle-server-1

# clresource create -Z zonecluster1 \
-g resource-group-1 \
-t SUNW.oracle_listener \
-p Oracle_home=/global/oracle \
-p Listener_name=ora-lsnr \
oracle-listener-1
```

▼ How to Register and Configure HA for Oracle Database With Oracle Grid Infrastructure for a Cluster (CLI)

This procedure explains the steps to register and configure HA for Oracle Database with clustered Oracle ASM instance using Oracle Solaris Cluster maintenance commands.

You can optionally use a third-party volume manager to provide candidate disks to clustered Oracle ASM disk groups. For this type of configuration, this procedure manually creates an Oracle Grid Infrastructure resource that proxies the Oracle Solaris Cluster `SUNW.ScalDeviceGroup` resource. In this procedure, the Oracle Grid Infrastructure resource is named `sun.`*resource*. You configure `sun.`*resource* to ensure that the corresponding Oracle ASM disk group is not mounted until `sun.`*resource* is online. The `sun.`*resource* resource comes online only if the corresponding `SUNW.ScalDeviceGroup` resource is online. And the `SUNW.ScalDeviceGroup` resource only comes online if the actual volume-manager disk set or disk group is online.

To ensure that the Oracle ASM disk group benefits from this dependency chain, after you define `sun.`*resource*, you modify the appropriate Oracle ASM disk-group resource so that the hard-start dependency includes `sun.`*resource*. Modifying the hard-start dependency of the Oracle ASM disk-group resource can only be performed by the `SUNW.scalable_asm_diskgroup_proxy` resource by using the VALIDATE method. Therefore, you must set an offline-restart dependency between the `SUNW.scalable_asm_diskgroup_proxy` and `SUNW.ScalDeviceGroup` resources.

Before You Begin

- Ensure that the `/etc/netmasks` file has IP-address subnet and netmask entries for all logical hostnames. If necessary, edit the `/etc/netmasks` file to add any missing entries.
- If you are using Solaris Volume Manager for Sun Cluster for volume management, configure a multi-owner disk set for clustered Oracle ASM to use. Follow procedures in "How to Create a Multi-Owner Disk Set in Solaris Volume Manager for Sun Cluster for the Oracle RAC Database" in "Oracle Solaris Cluster Data Service for Oracle Real Application Clusters Guide ".
- Ensure that Oracle Grid Infrastructure software is installed.

- Ensure that the Oracle Clusterware resource for the Oracle ASM instance and the database instance are configured.
- Ensure that the file system you will use for `Oracle_Home` is set up. See "Preparing the Oracle Solaris Cluster Nodes and Disks" on page 19 for instructions.

1. **On a cluster member, become superuser or assume a role that provides `solaris.cluster.modify` and `solaris.cluster.admin` RBAC authorizations.**

2. **If you are using a third-party volume manager or NFS as candidate disks for Oracle ASM disk groups, configure Oracle Grid Infrastructure.**

   Oracle ASM candidate disks can be derived from any of the following:

   - Solaris Volume Manager disk sets
   - NFS

   a. **Create the Oracle Grid Infrastructure `sun.storage_proxy.type` resource type.**

      ```
      # /Grid_home/bin/crsctl add type sun.storage_proxy.type -basetype local_resource
      ```

   b. **Create an Oracle Grid Infrastructure `sun.`*resource* resource of type `sun.storage_proxy.type`.**

Note - Ensure that all attribute values are enclosed in single quotes (`'`). Otherwise, the VALIDATE method of the `SUNW.scalable_asm_diskgroup_proxy` resource will fail the validation.

```
# /Grid_home/bin/crsctl add res sun.scal-asmdg1-rs \
      -type sun.storage_proxy.type \
      -attr "ACTION_SCRIPT='/opt/SUNWscor/dsconfig/bin/scproxy_crs_action' \
      ACL='owner:root:rwx,pgrp:oinstall:rwx,other::r--' SCRIPT_TIMEOUT='20'
 RESTART_ATTEMPTS='60'"
```

`sun.`*scal-asmdg1-rs*

The `SUNW.ScalDeviceGroup` resource name.

`-type sun.storage_proxy.type`

Specifies the `sun.storage_proxy.type` resource type.

`ACTION_SCRIPT`

Specifies the `/opt/SUNWscor/dsconfig/bin/scproxy_crs_action` action script.

`ACL`

Sets the owner equal to `root` and the group equal to the ACL group entry for the Oracle ASM disk group. The following command displays the ACL group entry:

```
# /Grid_home/bin/crsctl stat res ora.DATA1.dg -p | grep ACL=
ACL=owner:oragrid:rwx,pgrp:oinstall:rwx,other::r--
```

The example output shows that `oinstall` is the group entry.

`SCRIPT_TIMEOUT`

Set to `20`.

`RESTART_ATTEMPTS`

Set to `60`.

c. Verify that `sun.`*resource* is correctly defined.

Output is similar to the following:

```
# /Grid_home/bin/crsctl stat res sun.scal-asmdg1-rs -p
      NAME=sun.scal-asmdg1-rs
      TYPE=sun.storage_proxy.type
      ACL=owner:root:rwx,pgrp:oinstall:rwx,other::r--
      ACTIONS=
      ACTION_FAILURE_TEMPLATE=
      ACTION_SCRIPT=/opt/SUNWscor/dsconfig/bin/scproxy_crs_action
      ACTION_TIMEOUT=60
      AGENT_FILENAME=%CRS_HOME%/bin/scriptagent
      ALERT_TEMPLATE=
      ALIAS_NAME=
      AUTO_START=restore
      CHECK_INTERVAL=60
      CHECK_TIMEOUT=0
      CLEAN_TIMEOUT=60
      DEBUG=1
      DEFAULT_TEMPLATE=
      DEGREE=1
      DELETE_TIMEOUT=60
      DESCRIPTION=
      ENABLED=1
      INSTANCE_FAILOVER=1
      INTERMEDIATE_TIMEOUT=-
      LOAD=1
      LOGGING_LEVEL=1
      MODIFY_TIMEOUT=60
      NOT_RESTARTING_TEMPLATE=
      OFFLINE_CHECK_INTERVAL=0
      PROFILE_CHANGE_TEMPLATE=
      RESTART_ATTEMPTS=60
      SCRIPT_TIMEOUT=20
      SERVER_CATEGORY=
      START_CONCURRENCY=0
      START_DEPENDENCIES=
      START_TIMEOUT=0
      STATE_CHANGE_TEMPLATE=
```

```
      STOP_CONCURRENCEY=0
      STOP_DEPENDENCIES=
      STOP_TIMEOUT=0
      UPTIME_THRESHOLD=1h
      USER_WORKLOAD=no
#
```

d. Display the current offline-restart dependency.

```
# /Grid_home/bin/crsctl stat res ora.DATA1.dg -p | grep START_DEPENDENCIES
START_DEPENDENCIES=hard(ora.asm) pullup(ora.asm)

# clresource show -p Resource_dependencies_offline_restart asm-data1-rs

=== Resources ===

Resource: asm-data1-rs
Resource_dependencies_offline_restart: asm-inst-rs

-- Standard and extension properties --
```

e. Set the new dependency.

- **If *asm-data1-rs* already exists, use the following command to set the dependency.**

  Note that the command includes the plus (+) symbol:

  ```
  # clresource set -p Resource_dependencies_offline_restart+=scal-asmdg1-rs asm-data1-rs
  ```

  `-p Resource_dependencies_offline_restart+=`*resource*

  (For an Oracle ASM device group only) Sets the offline restart dependency for the specified Oracle ASM storage resource.

  asm-data1–rs

  Specifies the name of the resource that you are modifying.

- **If *asm-data1-rs* does not yet exist, use the following command to create the resource with the offline-restart dependency:**

  ```
  # clresource create -g asm-dg-rg \
  -t SUNW.scalable_asm_diskgroup_proxy \
  -p asm_diskgroups=data1 \
  -p Resource_dependencies_offline_restart=asm-inst-rs,
  scal-asmdg1-rs \
  -d asm-data1-rs
  ```

`-g` *asm-dg-rg*

Specifies the name of the Oracle ASM device group into which the resources are to be placed.

`-t` *resource-type*

Specifies the type of the resource to add.

`-p asm_diskgroups=`*data1*

Sets the name of the Oracle ASM disk group.

`-d`

Specifies disabling the resource that you create.

f. Verify the configured dependency.

```
# /Grid_home/bin/crsctl stat res ora.DATA1.dg -p | grep START_DEPENDENCIES
START_DEPENDENCIES=hard(ora.asm,sun.scal-asmdg1-rs) pullup(ora.asm)
# clresource show -p Resource_dependencies_offline_restart asm-data1-rs
=== Resources ===

Resource: asm-data1-rs
Resource_dependencies_offline_restart: asm-inst-rs scal-asmdg1-rs

-- Standard and extension properties --
```

3. Register the resource type for Oracle Clusterware framework.

Note - You can also use the `clsetup` utility to perform Step 3 through Step 5 in this procedure.

```
# clresourcetype register SUNW.crs_framework
```

4. Add a resource of type `SUNW.crs_framework` to the *rac-fmwk-rg* resource group.

```
# clresource create -g rac-fmwk-rg \
-t SUNW.crs_framework \
-p Resource_dependencies_offline_restart=rac-fmwk-rs \
-d crs-fmwk-rs
```

5. Create a failover resource group *ora-db-rg* for the Oracle database.

```
# clresourcegroup create ora-db-rg
```

6. If you are using a Sun QFS file system, perform the following steps to register the Sun QFS file system resource type and create the resource group.

a. Register the resource type for a Sun QFS file system.

```
# clresourcetype register SUNW.qfs
```

b. Create a resource group *qfs-rg*.

```
# clresourcegroup create qfs-rg
```

c. Add a resource of type `SUNW.qfs` to the *qfs-rg* resource group.

d. Add a resource of type `SUNW.qfs` to the *qfs-rg* resource group.

```
# clresource create -g qfs-rg -t SUNW.qfs -p QFSFileSystem=qfs-mp qfs-rs
```

`-g` *qfs-rg*

Specifies the name of the resource group.

`-t SUNW.qfs`

Specifies the type of the resource to add.

`-p QFSFileSystem=`*qfs-mp*

Specifies the Sun QFS file system mount point.

qfs-rs

Specifies the name of the Sun QFS file system resource that you are creating.

e. Bring online the *qfs-rg* resource group in a managed state on a cluster node.

```
# clresourcegroup online -eM qfs-rg
```

7. Register the `SUNW.HAStoragePlus` resource type and create the resource group.

a. Register the resource type for `SUNW.HAStoragePlus` resource type.

```
# clresourcetype register SUNW.HAStoragePlus
```

b. Add a resource of type `SUNW.HAStoragePlus` to the *ora-db-rg* resource group.

```
# clresource create -g ora-db-rg -t SUNW.HAStoragePlus \
-p filesystemmountpoints=mount-point-list -d hastp-rs
```

c. Bring online the *ora-db-rg* resource group in a managed state on a cluster node.

```
# clresourcegroup online -eM ora-db-rg
```

8. Register the Oracle ASM resource types for the data service.

a. Register the scalable ASM instance proxy resource type.

```
# clresourcetype register SUNW.scalable_asm_instance_proxy
```

b. Register the appropriate ASM disk-group resource type.

```
# clresourcetype register SUNW.scalable_asm_diskgroup_proxy
```

9. Create resource groups *asm-inst-rg* **and** *asm-dg-rg***.**

```
# clresourcegroup create -S asm-inst-rg asm-dg-rg
```

10. Set a strong positive affinity on *rac-fmwk-rg* **by** *asm-inst-rg***.**

```
# clresourcegroup set -p Rg_affinities=++rac-fmwk-rg asm-inst-rg
```

11. Set a strong positive affinity on *asm-inst-rg* **by** *asm-dg-rg***.**

```
# clresourcegroup set -p Rg_affinities=++asm-inst-rg asm-dg-rg
```

12. Add a resource of type `SUNW.crs_framework` to the *rac-fmwk-rg* **resource group.**

```
# clresource create -g rac-fmwk-rg \
-t SUNW.crs_framework \
-p Resource_dependencies_offline_restart=rac-fmwk-rs \
-d crs-fmwk-rs
```

13. If you configured the `$ORACLE_HOME` directory for Oracle ASM use on a cluster file system, configure resources to use that cluster file system.

Add resources of type `SUNW.HAStoragePlus` and `SUNW.scalable_asm_instance` to the *asm-inst-rg* resource group and set the appropriate dependencies between the two resources.

```
# clresource create -g asm-inst-rg -t SUNW.HAStoragePlus \
-p FilesystemMountPoints=cluster-file-system \
hastp-rs

# clresource create -g asm-inst-rg -t SUNW.scalable_asm_instance_proxy \
-p Oracle_home=Oracle_home \
-p CRS_HOME=Grid_home
-p Oracle_sid{node1}=instance \
-p Oracle_sid{node2}=instance \
-p Resource_dependencies_offline_restart=hastp-rs \
-p Resource_dependencies_offline_restart=crs-fmwk-rs \
asm-inst-rs
```

-g *asm-inst-rg*

Specifies the names of the resource groups into which the resources are to be placed.

`-p FilesystemMountPoints=`*cluster-file-system*

Specifies the name of the cluster file system.

hastp-rs

Specifies the name of the `SUNW.HAStoragePlus` resource to create.

`-p Oracle_home=`*Oracle_home*

Sets the path to the Oracle Database home directory.

`-p CRS_HOME=`*Grid_home*

Sets the path to the Oracle Grid Infrastructure for a Cluster home directory.

`-p "Oracle_sid{`*node*`}"=`*instance*

Sets the Oracle Database system identifier.

14. Add a resource of type `SUNW.scalable_asm_instance_proxy` to the *asm-inst-rg* resource group.

```
# clresource create -g asm-inst-rg \
-t SUNW.scalable_asm_instance_proxy \
-p Oracle_home=Oracle_home \
-p CRS_HOME=Grid_home \
-p "oracle_sid{node1}"=instance \
-p "oracle_sid{node2}"=instance \
-p Resource_dependencies_offline_restart=crs-fmwk-rs \
-d asm-inst-rs
```

`-t SUNW.asm_inst_proxy`

Specifies the type of the resource to add.

`-d` *asm-inst-rs*

Specifies the name of the resource that you are creating.

15. Add an ASM disk-group resource to the *asm-dg-rg* resource group.

Use the `SUNW.scalable_asm_diskgroup_proxy` resource type.

```
# clresource create -g asm-dg-rg \
-t SUNW.scalable_asm_diskgroup_proxy \
-p Asm_diskgroups=dg[,dg…] \
-p Resource_dependencies_offline_restart=asm-inst-rs,asm-stor-rs \
-d asm-dg-rs
```

16. Bring online the *asm-inst-rg* resource group in a managed state on a cluster node.

```
# clresourcegroup online -eM asm-inst-rg
```

17. **Bring online the *asm-dg-rg* resource group in a managed state on a cluster node.**

```
# clresourcegroup online -eM asm-dg-rg
```

18. **Verify the Oracle ASM installation by issuing the status command.**

```
# clresource status +
```

19. **Register the resource types for the HA for Oracle Database data service.**

You register two resource types, `SUNW.oracle_server` and `SUNW.oracle_listener`.

Note - Omit registration of the `SUNW.oracle_listener` resource type if you are using an Oracle Grid Infrastructure for Clusters Single Client Access Name (SCAN) listener.

```
# clresourcetype register SUNW.oracle_server
# clresourcetype register SUNW.oracle_listener
```

20. **Add a logical hostname resource to the failover resource group for Oracle Database.**

```
# clreslogicalhostname create -g ora-db-rg [-h logical-hostname] logical-hostname-rs
```

logical-hostname

Specifies a logical hostname. This logical hostname must be present in your name service database. If *logical-hostname* and *logical-hostname-rs* are identical, *logical-hostname* is optional.

logical-hostname-rs

Specifies the name that you are assigning to the logical hostname resource that you are creating.

21. **Bring online the failover resource group in a managed state on a cluster node.**

```
# clresourcegroup online -eM ora-db-rg
```

22. **Create an Oracle Database application server resource in the failover resource group.**

```
# clresource create -g ora-db-rg \
-t SUNW.oracle_server \
-p Db_unique_home=db-unique-home \
-p Resource_dependencies_offline_restart=asm-dg-rs \
-p Oracle_sid=instance \
-d ora-db-rs
```

-g *ora-db-rg*

Specifies the name of the resource group into which the resources are to be placed.

`-t SUNW.oracle_server`

Specifies the type of the resource to add.

`-p Oracle_sid=`*instance*

Sets the Oracle Database system identifier.

`-p Db_unique_name=`*db-unique-name*

Sets the Oracle Database unique name.

`-d` *ora-db-rs*

Specifies the name of the resource that you are creating.

Note - Optionally, you can set additional extension properties that belong to the HA for Oracle Database data service to override their default values. See "Setting HA for Oracle Database Extension Properties" on page 36 for a list of extension properties.

---

23. Bring online the Oracle Database server resource.

```
# clresource enable ora-db-rs
```

Next Steps Go to "Verifying the HA for Oracle Database Installation" on page 58 after you register and configure HA for Oracle Database.

Verifying the HA for Oracle Database Installation

Perform the following verification tests to make sure that you have correctly installed HA for Oracle Database.

These sanity checks ensure that all the Oracle Solaris Cluster nodes that run HA for Oracle Database can start the Oracle Database instance and that the other cluster nodes in the configuration can access the Oracle Database instance. Perform these sanity checks to isolate any problems in starting the Oracle Database software from HA for Oracle Database.

▼ How to Verify the HA for Oracle Database Installation

1. Log in as Oracle Database user `oracle` to the Oracle Solaris Cluster node that currently masters the Oracle Database resource group.

2. **Set the environment variables `ORACLE_SID` and `ORACLE_HOME`.**

3. **Confirm that you can start the Oracle Database instance from this cluster node.**

4. **Confirm that you can connect to the Oracle Database instance.**

   Use the `sqlplus` command with the `user/password` variable that is defined in the `connect_string` property.

   ```
   # sqlplus sysdba/passwd@tns_service
   ```

   tns_service

   Specifies the network name service provided by the `$ORACLE_HOME/network/admin/tnsnames.ora` file or the value of the `TNS_ADMIN` environment variable.

5. **Shut down the Oracle Database instance.**

   The Oracle Solaris Cluster software restarts the Oracle instance because the Oracle Database instance is under Oracle Solaris Cluster control.

6. **Switch the resource group that contains the Oracle Database resource to another cluster member.**

   ```
   # clresourcegroup switch -n node-zone-list resource-group
   ```

   resource-group

   Specifies the name of the resource group that you are switching.

7. **Log in as `oracle` to the cluster node that now contains the resource group.**

8. **Repeat Step 3 and Step 4 to confirm interactions with the Oracle Database instance.**

Oracle Database Clients

Clients must always refer to the database by using the network resource, not the physical hostname. The network resource is an IP address that can move between physical or virtual Oracle Solaris Cluster nodes during failover. The hostname is a physical or virtual machine name.

For example, in the `tnsnames.ora` file, you must specify the network resource as the host on which the database instance is running. See "How to Set Up Oracle Database Permissions" on page 31.

Note - Oracle Database client-server connections cannot survive an HA for Oracle Database switchover. The client application must be prepared to handle disconnection and reconnection or recovery as appropriate. A transaction monitor might simplify the application. Further, HA for Oracle Database cluster node recovery time is dependent on the application failover mechanism.

Location of HA for Oracle Database Log Files

Each instance of the HA for Oracle Database data service maintains log files in subdirectories of the `/var/opt/SUNWscor` directory.

- The `/var/opt/SUNWscor/oracle_server` directory contains log files for the Oracle Database server.
- The `/var/opt/SUNWscor/oracle_listener` directory contains log files for the Oracle Database listener.
- The `/var/opt/SUNWscor/oracle_asm` directory contains log file for Oracle ASM.

These files contain information about actions that the HA for Oracle Database data service performs. Refer to these files to obtain diagnostic information for troubleshooting your configuration or to monitor the behavior of the HA for Oracle Database data service.

Tuning the HA for Oracle Database Fault Monitors

Fault monitoring for the HA for Oracle Database data service is provided by the following fault monitors:

- The Oracle Database server fault monitor
- The Oracle Database listener fault monitor

Note - If you are using an Oracle Grid Infrastructure for Clusters Single Client Access Name (SCAN) listener, no fault monitoring is provided for the SCAN listener by Oracle Solaris Cluster software.

Each fault monitor is contained in a resource whose resource type is shown in the following table.

TABLE 1-3 Resource Types for HA for Oracle Database Fault Monitors

Fault Monitor	Resource Type
Oracle Database server	`SUNW.oracle_server`
Oracle Database listener	`SUNW.oracle_listener`

System properties and extension properties of these resources control the behavior of the fault monitors. The default values of these properties determine the preset behavior of the fault monitors. The preset behavior should be suitable for most Oracle Solaris Cluster installations. Therefore, you should tune the HA for Oracle Database fault monitors *only* if you need to modify this preset behavior.

Tuning the HA for Oracle Database fault monitors involves the following tasks:

- Setting the interval between fault monitor probes
- Setting the timeout for fault monitor probes
- Defining the criteria for persistent faults
- Specifying the failover behavior of a resource

For more information, see "Tuning Fault Monitors for Oracle Solaris Cluster Data Services" in "Oracle Solaris Cluster Data Services Planning and Administration Guide ". Information about the HA for Oracle Database fault monitors that you need to perform these tasks is provided in the subsections that follow.

Tune the HA for Oracle Database fault monitors when you register and configure HA for Oracle Database. For more information, see "Registering and Configuring HA for Oracle Database" on page 35.

Operation of the Oracle Database Server Fault Monitor

The fault monitor for the Oracle Database server uses a request to the server to query the health of the server.

The server fault monitor is started through `pmfadm` to make the monitor highly available. If the monitor is killed for any reason, the Process Monitor Facility (PMF) automatically restarts the monitor.

The server fault monitor consists of the following processes.

- A main fault monitor process
- A database client fault probe

This section contains the following information about the server fault monitor:

Operation of the Main Fault Monitor

The main fault monitor determines that an operation is successful if the database is online and no errors are returned during the transaction.

Operation of the Database Client Fault Probe

The database client fault probe performs the following operations:

1. Monitoring the partition for archived redo logs. See "Operations to Monitor the Partition for Archived Redo Logs" on page 62.
2. If the partition is healthy, determining whether the database is operational. See "Operations to Determine Whether the Database is Operational" on page 62.

The probe uses the timeout value that is set in the resource property `Probe_timeout` to determine how much time to allocate to successfully probe Oracle Database.

Operations to Monitor the Partition for Archived Redo Logs

The database client fault probe queries the dynamic performance view `v$archive_dest` to determine all possible destinations for archived redo logs. For every active destination, the probe determines whether the destination is healthy and has sufficient free space for storing archived redo logs.

- If the destination is healthy, the probe determines the amount of free space in the destination's file system. If the amount of free space is less than 10% of the file system's capacity and is less than 20 Mbytes, the probe prints a message to `syslog`.
- If the destination is in `ERROR` status, the probe prints a message to `syslog` and disables operations to determine whether the database is operational. The operations remain disabled until the error condition is cleared.

Operations to Determine Whether the Database is Operational

If the partition for archived redo logs is healthy, the database client fault probe queries the dynamic performance view `v$sysstat` to obtain database performance statistics. Changes to these statistics indicate that the database is operational. If these statistics remain unchanged between consecutive queries, the fault probe performs database transactions to determine if the database is operational. These transactions involve the creation, updating, and dropping of a table in the user table space.

The database client fault probe performs all its transactions as the Oracle Database user. The ID of this user is specified during the preparation of the Oracle Solaris Cluster nodes as explained in "How to Prepare the Oracle Solaris Cluster Nodes" on page 19.

Actions by the Server Fault Monitor in Response to a Database Transaction Failure

If a database transaction fails, the server fault monitor performs an action that is determined by the error that caused the failure. To change the action that the server fault monitor performs, customize the server fault monitor as explained in "Customizing the HA for Oracle Database Server Fault Monitor" on page 65.

If the action requires an external program to be run, the program is run as a separate process in the background.

Possible actions are as follows:

- **Ignore.** The server fault monitor ignores the error.
- **Stop monitoring.** The server fault monitor is stopped without shutting down the database.
- **Restart.** The server fault monitor stops and restarts the entity that is specified by the value of the `Restart_type` extension property:
  - If the `Restart_type` extension property is set to `RESOURCE_RESTART`, the server fault monitor restarts the database server resource. By default, the server fault monitor restarts the database server resource.
  - If the `Restart_type` extension property is set to `RESOURCE_GROUP_RESTART`, the server fault monitor restarts the database server resource group.

Note - The number of attempts to restart might exceed the value of the `Retry_count` resource property within the time that the `Retry_interval` resource property specifies. If this situation occurs, the server fault monitor attempts to switch over the resource group to another cluster node.

- **Switch over.** The server fault monitor switches over the database server resource group to another cluster node. If no nodes are available, the attempt to switch over the resource group fails. If the attempt to switch over the resource group fails, the database server is restarted.

Scanning of Logged Alerts by the Server Fault Monitor

Oracle Database logs alerts in an alert log file. The absolute path of this file is specified by the `alert_log_file` extension property of the `SUNW.oracle_server` resource. The server fault monitor scans the alert log file for new alerts at the following times:

- When the server fault monitor is started
- Each time that the server fault monitor queries the health of the server

If an action is defined for a logged alert that the server fault monitor detects, the server fault monitor performs the action in response to the alert.

Preset actions for logged alerts are listed in Table B-2. To change the action that the server fault monitor performs, customize the server fault monitor as explained in “Customizing the HA for Oracle Database Server Fault Monitor” on page 65.

Operation of the Oracle Database Listener Fault Monitor

The Oracle Database listener fault monitor checks the status of an Oracle Database listener.

If the listener is running, the Oracle Database listener fault monitor considers a probe successful. If the fault monitor detects an error, the listener is restarted.

Note - The listener resource does not provide a mechanism for setting the listener password. If Oracle Database listener security is enabled, a probe by the listener fault monitor might return Oracle Database error TNS-01169. Because the listener is able to respond, the listener fault monitor treats the probe as a success. This action does not cause a failure of the listener to remain undetected. A failure of the listener returns a different error, or causes the probe to time out.

The listener probe is started through `pmfadm` to make the probe highly available. If the probe is killed, PMF automatically restarts the probe.

If a problem occurs with the listener during a probe, the probe tries to restart the listener. The value that is set for the resource property `retry_count` determines the maximum number of times that the probe attempts the restart. If, after trying for the maximum number of times, the probe is still unsuccessful, the probe stops the fault monitor and does not switch over the resource group.

Obtaining Core Files for Troubleshooting DBMS Timeouts

To facilitate troubleshooting of unexplained DBMS timeouts, you can enable the fault monitor to create a core file when a probe timeout occurs. The contents of the core file relate to the fault monitor process. The fault monitor creates the core file in the root (`/`) directory. To enable the fault monitor to create a core file, use the `coreadm` command to enable set-id core dumps.

```
# coreadm -g /var/cores/%f.%n.%p.core -e global -e process \
-e global-setid -e proc-setid -e log
```

For more information, see the `coreadm`(1M) man page.

Customizing the HA for Oracle Database Server Fault Monitor

Customizing the HA for Oracle Database server fault monitor enables you to modify the behavior of the server fault monitor as follows:

- Overriding the preset action for an error
- Specifying an action for an error for which no action is preset

Caution - Before you customize the HA for Oracle Database server fault monitor, consider the effects of your customizations, especially if you change an action from restart or switch over to ignore or stop monitoring. If errors remain uncorrected for long periods, the errors might cause problems with the database. If you encounter problems with the database after customizing the HA for Oracle Database server fault monitor, revert to using the preset actions. Reverting to the preset actions enables you to determine if the problem is caused by your customizations.

Customizing the HA for Oracle Database server fault monitor involves the following activities:

1. Defining custom behavior for errors
2. Propagating a custom action file to all nodes in a cluster
3. Specifying the custom action file that a server fault monitor should use

Defining Custom Behavior for Errors

The HA for Oracle Database server fault monitor detects the following types of errors:

- DBMS errors that occur during a probe of the database by the server fault monitor
- Alerts that Oracle Database logs in the alert log file
- Timeouts that result from a failure to receive a response within the time that is set by the `Probe_timeout` extension property

To define custom behavior for these types of errors, create a custom action file. This section contains the following information about custom action files:

- “Custom Action File Format” on page 66
- “Changing the Response to a DBMS Error” on page 69
- “Changing the Response to Logged Alerts” on page 71
- “Changing the Maximum Number of Consecutive Timed-Out Probes” on page 72

Custom Action File Format

A custom action file is a plain text file. The file contains one or more entries that define the custom behavior of the HA for Oracle Database server fault monitor. Each entry defines the custom behavior for a single DBMS error, a single timeout error, or several logged alerts. A maximum of 1024 entries is allowed in a custom action file.

Note - Each entry in a custom action file overrides the preset action for an error, or specifies an action for an error for which no action is preset. Create entries in a custom action file *only* for the preset actions that you are overriding or for errors for which no action is preset. Do *not* create entries for actions that you are not changing.

An entry in a custom action file consists of a sequence of keyword-value pairs that are separated by semicolons. Each entry is enclosed in braces.

The format of an entry in a custom action file is as follows:

```
{
[ERROR_TYPE=DBMS_ERROR|SCAN_LOG|TIMEOUT_ERROR;]
ERROR=error-spec;
[ACTION=SWITCH|

RESTART|STOP|NONE;]
[CONNECTION_STATE=co|di|on|*;]
[NEW_STATE=co|di|on|*;]
[MESSAGE="message-string"]
}
```

White space may be used between separated keyword-value pairs and between entries to format the file.

The meaning and permitted values of the keywords in a custom action file are as follows:

`ERROR_TYPE`

Indicates the type of the error that the server fault monitor has detected. The following values are permitted for this keyword:

`DBMS_ERROR`

Specifies that the error is a DBMS error.

`SCAN_LOG`

Specifies that the error is an alert that is logged in the alert log file.

`TIMEOUT_ERROR`

Specifies that the error is a timeout.

The ERROR_TYPE keyword is optional. If you omit this keyword, the error is assumed to be a DBMS error.

ERROR

Identifies the error. The data type and the meaning of *error-spec* are determined by the value of the ERROR_TYPE keyword as shown in the following table.

ERROR_TYPE	Data Type	Meaning
DBMS_ERROR	Integer	The error number of a DBMS error that is generated by Oracle Database
SCAN_LOG	Quoted regular expression	A string in an error message that Oracle Database has logged to the Oracle Database alert log file
TIMEOUT_ERROR	Integer	The number of consecutive timed-out probes since the server fault monitor was last started or restarted

You must specify the ERROR keyword. If you omit this keyword, the entry in the custom action file is ignored.

ACTION

Specifies the action that the server fault monitor is to perform in response to the error. The following values are permitted for this keyword:

NONE

Specifies that the server fault monitor ignores the error.

STOP

Specifies that the server fault monitor is stopped.

RESTART

Specifies that the server fault monitor stops and restarts the entity that is specified by the value of the Restart_type extension property of the SUNW.oracle_server resource.

SWITCH

Specifies that the server fault monitor switches over the database server resource group to another cluster node.

The ACTION keyword is optional. If you omit this keyword, the server fault monitor ignores the error.

CONNECTION_STATE

Specifies the required state of the connection between the database and the server fault monitor when the error is detected. The entry applies only if the connection is in the

required state when the error is detected. The following values are permitted for this keyword:

`*`

Specifies that the entry always applies, regardless of the state of the connection.

`co`

Specifies that the entry applies only if the server fault monitor is attempting to connect to the database.

`on`

Specifies that the entry applies only if the server fault monitor is online. The server fault monitor is online if it is connected to the database.

`di`

Specifies that the entry applies only if the server fault monitor is disconnecting from the database.

The `CONNECTION_STATE` keyword is optional. If you omit this keyword, the entry always applies, regardless of the state of the connection.

`NEW_STATE`

Specifies the state of the connection between the database and the server fault monitor that the server fault monitor must attain after the error is detected. The following values are permitted for this keyword:

`*`

Specifies that the state of the connection must remain unchanged.

`co`

Specifies that the server fault monitor must disconnect from the database and reconnect immediately to the database.

`di`

Specifies that the server fault monitor must disconnect from the database. The server fault monitor reconnects when it next probes the database.

The `NEW_STATE` keyword is optional. If you omit this keyword, the state of the database connection remains unchanged after the error is detected.

`MESSAGE`

Specifies an additional message that is printed to the resource's log file when this error is detected. The message must be enclosed in double quotes. This message is additional to the standard message that is defined for the error.

The `MESSAGE` keyword is optional. If you omit this keyword, no additional message is printed to the resource's log file when this error is detected.

Changing the Response to a DBMS Error

The action that the server fault monitor performs in response to each DBMS error is preset as listed in Table B-1. To determine whether you need to change the response to a DBMS error, consider the effect of DBMS errors on your database to determine if the preset actions are appropriate. For examples, see the subsections that follow:

- "Responding to an Error Whose Effects Are Major" on page 69
- "Ignoring an Error Whose Effects Are Minor" on page 70

To change the response to a DBMS error, create an entry in a custom action file in which the keywords are set as follows:

- `ERROR_TYPE` is set to `DBMS_ERROR`.
- `ERROR` is set to the error number of the DBMS error.
- `ACTION` is set to the action that you require.

Responding to an Error Whose Effects Are Major

If an error that the server fault monitor ignores affects more than one session, action by the server fault monitor might be required to prevent a loss of service.

For example, no action is preset for Oracle Database error 4031: `unable to allocate` *num-bytes* `bytes of shared memory`. However, this Oracle Database error indicates that the shared global area (SGA) has insufficient memory, is badly fragmented, or both states apply. If this error affects only a single session, ignoring the error might be appropriate. However, if this error affects more than one session, consider specifying that the server fault monitor restart the database.

The following example shows an entry in a custom action file for changing the response to a DBMS error to restart.

EXAMPLE 1-3 Changing the Response to a DBMS Error to Restart

```
{
ERROR_TYPE=DBMS_ERROR;
ERROR=4031;
ACTION=restart;
CONNECTION_STATE=*;
NEW_STATE=*;
MESSAGE="Insufficient memory in shared pool.";
}
```

This example shows an entry in a custom action file that overrides the preset action for DBMS error 4031. This entry specifies the following behavior:

- In response to DBMS error 4031, the action that the server fault monitor performs is restart.
- This entry applies regardless of the state of the connection between the database and the server fault monitor when the error is detected.
- The state of the connection between the database and the server fault monitor must remain unchanged after the error is detected.
- The following message is printed to the resource's log file when this error is detected:

```
Insufficient memory in shared pool.
```

Ignoring an Error Whose Effects Are Minor

If the effects of an error to which the server fault monitor responds are minor, ignoring the error might be less disruptive than responding to the error.

For example, the preset action for Oracle Database error 4030: `out of process memory when trying to allocate` *num-bytes* `bytes` is restart. This Oracle Database error indicates that the server fault monitor could not allocate private heap memory. One possible cause of this error is that insufficient memory is available to the operating system. If this error affects more than one session, restarting the database might be appropriate. However, this error might not affect other sessions because these sessions do not require further private memory. In this situation, consider specifying that the server fault monitor ignore the error.

The following example shows an entry in a custom action file for ignoring a DBMS error.

EXAMPLE 1-4 Ignoring a DBMS Error

```
{
ERROR_TYPE=DBMS_ERROR;
ERROR=4030;
ACTION=none;
CONNECTION_STATE=*;
NEW_STATE=*;
MESSAGE="";
}
```

This example shows an entry in a custom action file that overrides the preset action for DBMS error 4030. This entry specifies the following behavior:

- The server fault monitor ignores DBMS error 4030.
- This entry applies regardless of the state of the connection between the database and the server fault monitor when the error is detected.
- The state of the connection between the database and the server fault monitor must remain unchanged after the error is detected.
- No additional message is printed to the resource's log file when this error is detected.

Changing the Response to Logged Alerts

Oracle Database software logs alerts in a file that is identified by the `alert_log_file` extension property. The server fault monitor scans this file and performs actions in response to alerts for which an action is defined.

Logged alerts for which an action is preset are listed in Table B-2. Change the response to logged alerts to change the preset action, or to define new alerts to which the server fault monitor responds.

To change the response to logged alerts, create an entry in a custom action file in which the keywords are set as follows:

- `ERROR_TYPE` is set to `SCAN_LOG`.
- `ERROR` is set to a quoted regular expression that identifies a string in an error message that Oracle Database has logged to the Oracle Database alert log file.
- `ACTION` is set to the action that you require.

The server fault monitor processes the entries in a custom action file in the order in which the entries occur. Only the first entry that matches a logged alert is processed. Later entries that match are ignored. If you are using regular expressions to specify actions for several logged alerts, ensure that more specific entries occur before more general entries. Specific entries that occur after general entries might be ignored.

For example, a custom action file might define different actions for errors that are identified by the regular expressions `ORA-65` and `ORA-6`. To ensure that the entry that contains the regular expression `ORA-65` is not ignored, ensure that this entry occurs before the entry that contains the regular expression `ORA-6`.

The following example shows an entry in a custom action file for changing the response to a logged alert.

EXAMPLE 1-5 Changing the Response to a Logged Alert

```
{
ERROR_TYPE=SCAN_LOG;
ERROR="ORA-00600: internal error";
ACTION=RESTART;
}
```

This example shows an entry in a custom action file that overrides the preset action for logged alerts about internal errors. This entry specifies the following behavior:

- In response to logged alerts that contain the text `ORA-00600: internal error`, the action that the server fault monitor performs is restart.

- This entry applies regardless of the state of the connection between the database and the server fault monitor when the error is detected.
- The state of the connection between the database and the server fault monitor must remain unchanged after the error is detected.
- No additional message is printed to the resource's log file when this error is detected.

Changing the Maximum Number of Consecutive Timed-Out Probes

By default, the server fault monitor restarts the database after the second consecutive timed-out probe. If the database is lightly loaded, two consecutive timed-out probes should be sufficient to indicate that the database is hanging. However, during periods of heavy load, a server fault monitor probe might time out even if the database is functioning correctly. To prevent the server fault monitor from restarting the database unnecessarily, increase the maximum number of consecutive timed-out probes.

Caution - Increasing the maximum number of consecutive timed-out probes increases the time that is required to detect that the database is hanging.

To change the maximum number of consecutive timed-out probes allowed, create one entry in a custom action file for each consecutive timed-out probe that is allowed *except* the first timed-out probe.

Note - You are not required to create an entry for the first timed-out probe. The action that the server fault monitor performs in response to the first timed-out probe is preset.

For the last allowed timed-out probe, create an entry in which the keywords are set as follows:

- `ERROR_TYPE` is set to `TIMEOUT_ERROR`.
- `ERROR` is set to the maximum number of consecutive timed-out probes that are allowed.
- `ACTION` is set to `RESTART`.

For each remaining consecutive timed-out probe except the first timed-out probe, create an entry in which the keywords are set as follows:

- `ERROR_TYPE` is set to `TIMEOUT_ERROR`.
- `ERROR` is set to the sequence number of the timed-out probe. For example, for the second consecutive timed-out probe, set this keyword to 2. For the third consecutive timed-out probe, set this keyword to 3.
- `ACTION` is set to `NONE`.

Tip - To facilitate debugging, specify a message that indicates the sequence number of the timed-out probe.

The following example shows the entries in a custom action file for increasing the maximum number of consecutive timed-out probes to five.

EXAMPLE 1-6 Changing the Maximum Number of Consecutive Timed-Out Probes

```
{
ERROR_TYPE=TIMEOUT;
ERROR=2;
ACTION=NONE;
CONNECTION_STATE=*;
NEW_STATE=*;
MESSAGE="Timeout #2 has occurred.";
}

{
ERROR_TYPE=TIMEOUT;
ERROR=3;
ACTION=NONE;
CONNECTION_STATE=*;
NEW_STATE=*;
MESSAGE="Timeout #3 has occurred.";
}

{
ERROR_TYPE=TIMEOUT;
ERROR=4;
ACTION=NONE;
CONNECTION_STATE=*;
NEW_STATE=*;
MESSAGE="Timeout #4 has occurred.";
}

{
ERROR_TYPE=TIMEOUT;
ERROR=5;
ACTION=RESTART;
CONNECTION_STATE=*;
NEW_STATE=*;
MESSAGE="Timeout #5 has occurred. Restarting.";
}
```

This example shows the entries in a custom action file for increasing the maximum number of consecutive timed-out probes to five. These entries specify the following behavior:

- The server fault monitor ignores the second consecutive timed-out probe through the fourth consecutive timed-out probe.
- In response to the fifth consecutive timed-out probe, the action that the server fault monitor performs is restart.
- The entries apply regardless of the state of the connection between the database and the server fault monitor when the timeout occurs.

- The state of the connection between the database and the server fault monitor must remain unchanged after the timeout occurs.
- When the second consecutive timed-out probe through the fourth consecutive timed-out probe occurs, a message of the following form is printed to the resource's log file:

  `Timeout #`*number* `has occurred.`
- When the fifth consecutive timed-out probe occurs, the following message is printed to the resource's log file:

  ```
  Timeout #5 has occurred. Restarting.
  ```

Propagating a Custom Action File to All Nodes in a Cluster

A server fault monitor must behave consistently on all cluster nodes. Therefore, the custom action file that the server fault monitor uses must be identical on all cluster nodes. After creating or modifying a custom action file, ensure that this file is identical on all cluster nodes by propagating the file to all cluster nodes. To propagate the file to all cluster nodes, use the method that is most appropriate for your cluster configuration:

- Locating the file on a file system that all cluster nodes share
- Locating the file on a highly available local file system
- Copying the file to the local file system of each cluster node by using operating system commands such as the `rcp` command or the `rdist` command.

Specifying the Custom Action File That a Server Fault Monitor Should Use

To apply customized actions to a server fault monitor, you must specify the custom action file that the fault monitor should use. Customized actions are applied to a server fault monitor when the server fault monitor reads a custom action file. A server fault monitor reads a custom action file when the you specify the file.

Specifying a custom action file also validates the file. If the file contains syntax errors, an error message is displayed. Therefore, after modifying a custom action file, specify the file again to validate the file.

Caution - If syntax errors in a modified custom action file are detected, correct the errors before the fault monitor is restarted. If the syntax errors remain uncorrected when the fault monitor is restarted, the fault monitor reads the erroneous file, ignoring entries that occur after the first syntax error.

▼ How to Specify the Custom Action File That a Server Fault Monitor Should Use

1. **On a cluster node, become superuser or assume a role that provides `solaris.cluster.modify` RBAC authorization.**

2. **Set the `Custom_action_file` extension property of the `SUNW.oracle_server` resource.**

   Set this property to the absolute path of the custom action file.

   ```
   # clresource set -p custom_action_file=filepath  server-resource
   ```

   `-p custom_action_file=`*filepath*

   Specifies the absolute path of the custom action file.

   server-resource

   Specifies the `SUNW.oracle_server` resource.

Upgrading HA for Oracle Database Resource Types

This section provides the following information about upgrading HA for Oracle Database resource types:

Overview of Upgrading HA for Oracle Database Resource Types

The existing resource types for the HA for Oracle Database data service are as follows:

- `SUNW.oracle_listener`, which represents an Oracle Database listener
- `SUNW.oracle_server`, which represents an Oracle Database server

Note - If you are using an Oracle Grid Infrastructure for Clusters Single Client Access Name (SCAN) listener, the `SUNW.oracle_listener` resource type is not used.

In addition to the existing resource types, there are three additional resource types included in HA for Oracle Database data service for Oracle ASM. These resource types are as follows:

- `SUNW.scalable_asm_diskgroup_proxy`, which represents the single or clustered instance Oracle ASM disk group in an Oracle Solaris Cluster configuration. For more information about the resource type, see the `SUNW.scalable_asm_diskgroup_proxy`(5) man page.
- `SUNW.scalable_asm_instance`, which represents the single instance Oracle ASM in an Oracle Solaris Cluster configuration. For more information about the resource type, see the `SUNW.scalable_asm_instance`(5) man page.

Note - Single instance Oracle ASM is not supported in an Oracle Solaris Cluster 4.2 configuration.

- `SUNW.scalable_asm_instance_proxy`, which represents a proxy for the clustered Oracle ASM instance in an Oracle Solaris Cluster configuration. For more information about the resource type, see the `SUNW.scalable_asm_instance_proxy`(5) man page.

Upgrade the existing resource types if you are upgrading from an earlier version of HA for Oracle Database.

For general instructions that explain how to upgrade a resource type, see "Upgrading a Resource Type" in "Oracle Solaris Cluster Data Services Planning and Administration Guide ".

Upgrading the `SUNW.oracle_listener` Resource Type

The information that you require to complete the upgrade of the `SUNW.oracle_listener` resource type is provided in the following subsections:

- "Information for Registering the New `SUNW.oracle_listener` Resource Type Version" on page 76
- "Information for Migrating Existing Instances of the `SUNW.oracle_listener` Resource Type" on page 77

Information for Registering the New `SUNW.oracle_listener` Resource Type Version

To determine the version of the resource type that is currently registered, use one command from the following list:

- `clresourcetype list`
- `clresourcetype show`

The resource type registration (RTR) file for this resource type is `/SUNW.oracle_listener`.

To determine the version of the latest installed version of the `/opt/cluster/lib/rgm/rtreg/SUNW.oracle_listener` resource type, whether it is registered or not, use the following command:

```
# grep -i RT_VERSION /opt/cluster/lib/rgm/rtreg/SUNW.oracle_listener
```

If the version of the latest installed resource type is later than the registered version, migrate to the newer version to ensure full functionality.

Note - In the global zone, the Resource Group option of `clsetup` locates for you the available resource type versions you can upgrade to.

Information for Migrating Existing Instances of the `SUNW.oracle_listener` Resource Type

The information that you require to edit each instance of the `SUNW.oracle_listener` resource type is as follows:

- You can perform the migration at any time.
- If you need to specify the timeout value in seconds that the fault monitor uses to probe an Oracle Database listener, set the `Probe_timeout` extension property. For more information, see "`SUNW.oracle_listener` Extension Properties" on page 85.

The following example shows a command for editing an instance of the `SUNW.oracle_listener` resource type.

EXAMPLE 1-7 Editing an Instance of the `SUNW.oracle_listener` Resource Type

```
# clresource set -p Type_version=N \
-p probe_timeout=60 oracle-lrs
```

This command edits a `SUNW.oracle_listener` resource as follows:

- The `SUNW.oracle_listener` resource is named `oracle-lrs`.
- The `Type_version` property of this resource is set to *N*, which represents the version number of the resource type you migrated to.
- The timeout value in seconds that the fault monitor uses to probe an Oracle Database listener is set to 60 seconds.

Upgrading the SUNW.oracle_server Resource Type

The information that you require to complete the upgrade of the SUNW.oracle_server resource type is provided in the following subsections:

- "Information for Registering the New SUNW.oracle_server Resource Type Version" on page 78
- "Information for Migrating Existing Instances of the SUNW.oracle_server Resource Type" on page 78

Information for Registering the New SUNW.oracle_server Resource Type Version

To determine the version of the resource type that is registered, use one command from the following list:

- clresourcetype list
- clresourcetype show

The resource type registration (RTR) file for this resource type is /opt/SUNWscor/oracle_server/etc/SUNW.oracle_server.

To determine the version of the latest installed version of the SUNW.oracle_server resource type, whether it is registered or not, use the following command:

```
# grep -i RT_VERSION /opt/cluster/lib/rgm/rtreg/SUNW.oracle_server
```

If the version of the latest installed resource type is later than the registered version, migrate to the newer version to ensure full functionality.

Note - The Resource Group option of clsetup locates for you the available resource type versions you can upgrade to.

Information for Migrating Existing Instances of the SUNW.oracle_server Resource Type

The information that you require to edit each instance of the SUNW.oracle_server resource type is as follows:

- You can perform the migration at any time.

- If you customized the behavior of the server fault monitor, set the `Custom_action_file` extension property. For more information, see "Customizing the HA for Oracle Database Server Fault Monitor" on page 65.

The following example shows a command for editing an instance of the `SUNW.oracle_server` resource type.

EXAMPLE 1-8 Editing an Instance of the `SUNW.oracle_server` Resource Type

```
# clresource set  -p Type_version=N \
-p custom_action_file=/opt/SUNWscor/oracle_server/etc/srv_mon_cust_actions \
oracle-srs
```

This command edits a `SUNW.oracle_server` resource as follows:

- The `SUNW.oracle_server` resource is named `oracle-srs`.
- The `Type_version` property of this resource is set to *N*, which represents the version number of the resource type you migrated to.
- Custom behavior for the fault monitor of this resource is specified in the file `/opt/SUNWscor/oracle_server/etc/srv_mon_cust_actions`.

Changing the Role of an Oracle Data Guard Instance

Database role failover or switchover is possible between an Oracle Database primary database and an Oracle Database standby database. When you use Oracle Database commands to change the role of Oracle Data Guard instances, the changes are not propagated to the Oracle Solaris Cluster resources that represent these instances. Therefore, you must also use Oracle Solaris Cluster commands to change extension properties of these resources to ensure that database instances are started in the correct role.

▼ How to Change the Role of an Oracle Data Guard Instance

1. **Prevent Oracle Solaris Cluster from starting the instance in an incorrect role.**

   If an Oracle Solaris Cluster node fails while you are changing the role of an Oracle Data Guard instance, Oracle Solaris Cluster software might restart the instance in an incorrect role. To prevent this possibility, change the `Dataguard_role` extension property of the Oracle Database server resource that represents the instance to `IN_TRANSITION`.

```
# clresource set -p Dataguard_role=IN_TRANSITION server-rs
```

2. **Perform the required operations on the Oracle database to convert the database to a new role.**

3. **Change the following extension properties of the Oracle Database server resource that represents the instance to reflect the new role of the instance:**

   - `Dataguard_role`
   - `Standby_mode`

   The required combination of `Dataguard_role` and `Standby_mode` depends on the change of role, as follows:

   - **To change from a primary database to a physical standby database, run the following command:**

     ```
     # clresource set -p Dataguard_role=STANDBY -p Standby_mode=PHYSICAL server-rs
     ```

   - **To change from a primary database to a logical standby database, run the following command:**

     ```
     # clresource set -p Dataguard_role=STANDBY \
     -p Standby_mode=LOGICAL server-rs
     ```

   - **To change from a standby database to a primary database, run the following command:**

     ```
     # clresource set -p Dataguard_role=PRIMARY server-rs
     ```

   - **To change from a physical standby database to a snapshot standby database, run the following command:**

     ```
     # clresource set -p Standby_mode=SNAPSHOT server-rs
     ```

♦ ♦ ♦ APPENDIX A

HA for Oracle Database Extension Properties

The extension properties that you can set for each HA for Oracle Database resource type are listed in the following sections:

See the `r_properties`(5) man page and the `rg_properties`(5) man page for details about all of the system-defined properties.

`SUNW.oracle_server` Extension Properties

`Auto_End_Bkp` (Boolean)

Specifies whether the following recovery actions are performed if an Oracle Database relational database management system (RDBMS) hot backup is interrupted.

- Recognizing when a database fails to open because of files that remain in hot backup mode. This verification process occurs when HA for Oracle Database starts.
- Identifying and releasing all files that remain in hot backup mode.
- Opening the database for use.

The permitted values for this property are as follows:

- `False` – Specifies that the recovery actions are *not* performed. This value is the default.
- `True` – Specifies that the recovery actions are performed.

Default: False

Range: None

Tunable: Any time

`Connect_cycle` (integer)

The number of probe cycles that the server fault monitor performs before disconnecting from the database.

Default: 5

Range: 0 – 99,999

Tunable: Any time

`Custom_action_file` (string)

The absolute path of the file that defines the custom behavior of the HA for Oracle Database server fault monitor.

Default: “”

Range: None

Tunable: Any time

`Dataguard_role` (string)

The role of the database. The permitted values for this property are as follows:

`NONE`

Specifies that no standby database instances configured for the database instance

`PRIMARY`

Specifies that the database is a primary database instance for which standby database instances are configured

`STANDBY`

Specifies that the database role is standby

`IN_TRANSITION`

Specifies that the database is undergoing a role reversal process

Default: `NONE`

Range: None

Tunable: Any time

`Db_unique_name` (string)

The unique name of the single-instance Oracle database that is being deployed.

Default: `NONE`

Range: None

Tunable: Any time

`Debug_level` (integer)

The level to which debug messages from the Oracle Database server component are logged. When the debug level is increased, more debug messages are written to the log files. These messages are logged to the file `/var/opt/SUNWscor/oracle_server/message_log.`*rs*, where *rs* is the name of the resource that represents the Oracle Database server component.

Default: 1, which logs `syslog` messages

Range: 0– 100

Tunable: Any time

`Oracle_home` (string)

The path to the Oracle Database home directory.

Default: None

Range: Minimum = 1

Tunable: When disabled

`Oracle_sid` (string)

The Oracle Database system identifier.

Default: None

Range: Minimum = 1

Tunable: When disabled

`Parameter_file` (string)

The Oracle Database parameter file. If the Oracle Database parameter file is not specified, this property defaults to the Oracle Database default.

Default: ““

Range: Minimum = 0

Tunable: Any time

`Probe_timeout` (integer)

The timeout value (in seconds) that the server fault monitor uses to probe an Oracle Database server instance.

Default: 300

Range: 0– 99,999

Tunable: Any time

`Restart_type` (string)

Specifies the entity that the server fault monitor restarts when the response to a fault is restart. The permitted values for this property are as follows:

`RESOURCE_RESTART`

Specifies that only this resource is restarted

`RESOURCE_GROUP_RESTART`

Specifies that all resources in the resource group that contains this resource are restarted

Default: RESOURCE_RESTART

Range: None

Tunable: Any time

Standby_mode (string)

The mode of the standby database. The permitted values for this property are as follows:

LOGICAL

Specifies a logical standby database

PHYSICAL

Specifies a physical standby database

SNAPSHOT

Specifies a snapshot standby database

Default: LOGICAL

Range: None

Tunable: Any time

User_env (string)

A file that contains environment variables to be set before server startup and shutdown. Those environment variables that have values that differ from Oracle Database defaults must be defined in this file.

For example, a user's listener.ora file might not reside under the /var/opt/oracle directory or the $ORACLE_HOME/network/admin. directory. In this situation, the TNS_ADMIN environment variable should be defined.

The definition of each environment variable that is defined must follow the format VARIABLE_NAME=VARIABLE_VALUE. Each of these environment variables must be specified, one per line in the environment file.

Default: NULL

Range: None

Tunable: Any time

Wait_for_online (Boolean)

Wait in the START method until the database is online.

Default: True

Range: None

Tunable: Any time

SUNW.oracle_listener Extension Properties

`Listener_name` (string)

The name of the Oracle Database listener. This name must match the corresponding entry in the `listener.ora` configuration file.

Default: `LISTENER`

Range: Not applicable

Tunable: When disabled

`Oracle_home` (string)

The path to the Oracle Database home directory.

Default: No default defined

Range: Not applicable

Tunable: When disabled

`Probe_timeout` (integer)

The timeout value in seconds that the fault monitor uses to probe an Oracle Database listener.

Default: 180

Range: 1– 99,999

Tunable: Any time

`User_env` (string)

A file that contains environment variables to be set before listener startup and shutdown. Those environment variables that have values that differ from Oracle Database defaults must be defined in this file.

For example, a user's `listener.ora` file might not reside under the `/var/opt/oracle` directory or the `$ORACLE_HOME/network/admin`. directory. In this situation, the `TNS_ADMIN` environment variable should be defined.

The definition of each environment variable that is defined must follow the format `VARIABLE_NAME=VARIABLE_VALUE`. Each of these environment variables must be specified, one per line in the environment file.

Default: ““

Range: Not applicable

Tunable: Any time

♦ ♦ ♦ APPENDIX B

Preset Actions for DBMS Errors and Logged Alerts

Preset actions for DBMS errors and logged alerts are listed as follows:

- DBMS errors for which an action is preset are listed in Table B-1.
- Logged alerts for which an action is preset are listed in Table B-2.

TABLE B-1 Preset Actions for DBMS Errors

Error Number	Action	Connection State	New State	Message
18	NONE	co	di	Max. number of DBMS sessions exceeded
20	NONE	co	di	Max. number of DBMS processes exceeded
28	NONE	on	di	Session killed by DBA, will reconnect
50	SWITCH	*	di	O/S error occurred while obtaining an enqueue. See o/s error.
51	NONE	*	di	timeout occurred while waiting for resource
55	NONE	*	*	maximum number of DML locks in DBMS exceeded
62	STOP	*	di	Need to set DML_LOCKS in init.ora file to value other than 0
107	RESTART	*	di	failed to connect to ORACLE listener process
257	NONE	*	di	archiver error. Connect internal only, until freed.
290	SWITCH	*	di	Operating system archival error occurred. Check alert log.
447	SWITCH	*	di	fatal error in background process
448	RESTART	*	di	normal completion of background process
449	RESTART	*	di	background process `%s' unexpectedly terminated with error %s
470	SWITCH	*	di	Oracle background process died
471	SWITCH	*	di	Oracle background process died
472	SWITCH	*	di	Oracle background process died
473	SWITCH	*	di	Oracle background process died
474	RESTART	*	di	SMON died, warm start required
475	SWITCH	*	di	Oracle background process died
476	SWITCH	*	di	Oracle background process died

Error Number	Action	Connection State	New State	Message
477	SWITCH	*	di	Oracle background process died
480	RESTART	*	di	LCK* process terminated with error
481	RESTART	*	di	LMON process terminated with error
482	RESTART	*	di	LMD* process terminated with error
602	SWITCH	*	di	internal programming exception
604	NONE	on	di	Recursive error
705	RESTART	*	di	inconsistent state during start up
942	NONE	on	*	Warning - V$SYSSTAT not accessible - check grant on V_$SYSSTAT
1001	NONE	on	di	Lost connection to database
1002	NONE	on	*	Internal error in HA-DBMS Oracle
1003	NONE	on	di	Resetting database connection
1012	NONE	on	di	Not logged on
1012	RESTART	di	co	Not logged on
1014	NONE	*	*	ORACLE shutdown in progress
1017	STOP	*	*	Please correct login information in HA-DBMS Oracle database configuration
1031	NONE	on	*	Insufficient privileges to perform DBMS operations - check Oracle user privileges
1033	NONE	co	co	Oracle is in the shutdown or initialization process
1033	NONE	*	di	Oracle is in the shutdown or initialization process
1034	RESTART	co	co	Oracle is not available
1034	RESTART	di	co	Oracle is not available
1034	NONE	on	di	Oracle is not available
1035	RESTART	co	co	Access restricted - restarting database to reset
1041	NONE	on	di	
1041	NONE	di	co	
1045	NONE	co	*	Fault monitor user lacks CREATE SESSION privilege logon denied.
1046	RESTART	*	di	cannot acquire space to extend context area
1050	RESTART	*	di	cannot acquire space to open context area
1053	SWITCH	*	*	user storage address cannot be read or written
1054	SWITCH	*	*	user storage address cannot be read or written
1075	NONE	co	on	Already logged on
1089	NONE	on	di	immediate shutdown in progresss
1089	NONE	*	*	Investigate! Could be hanging!
1090	NONE	*	di	shutdown in progress - connection is not permitted
1092	NONE	*	di	ORACLE instance terminated. Disconnection forced
1513	SWITCH	*	*	invalid current time returned by operating system

Error Number	Action	Connection State	New State	Message
1542	NONE	on	*	table space is off-line - please correct!
1552	NONE	on	*	rollback segment is off-line - please correct!
1950	NONE	on	*	Insufficient privileges to perform DBMS operations - check Oracle user privileges
2701	STOP	*	*	HA-DBMS Oracle error - ORACLE_HOME did not get set!
2703	RESTART	*	di	
2704	RESTART	*	di	
2709	RESTART	*	di	
2710	RESTART	*	di	
2719	RESTART	*	di	
2721	RESTART	*	*	
2726	STOP	*	*	Could not locate ORACLE executables - check ORACLE_HOME setting
2735	RESTART	*	*	osnfpm: cannot create shared memory segment
2811	SWITCH	*	*	Unable to attach shared memory segment
2839	SWITCH	*	*	Sync of blocks to disk failed.
2840	SWITCH	*	*	
2846	SWITCH	*	*	
2847	SWITCH	*	*	
2849	SWITCH	*	*	
2842	RESTART	*	*	Client unable to fork a server - Out of memory
3113	RESTART	co	di	lost connection
3113	NONE	on	di	lost connection
3113	NONE	di	di	lost connection
3114	NONE	*	co	Not connected?
4030	RESTART	*	*	
4032	RESTART	*	*	
4100	RESTART	*	*	communication area cannot be allocated insufficient memory
6108	STOP	co	*	Can't connect to remote database - make sure SQL*Net server is up
6114	STOP	co	*	Can't connect to remote database - check SQL*Net configuration
7205	SWITCH	*	di	
7206	SWITCH	*	di	
7208	SWITCH	*	di	
7210	SWITCH	*	di	
7211	SWITCH	*	di	
7212	SWITCH	*	di	
7213	SWITCH	*	di	

Error Number	Action	Connection State	New State	Message
7214	SWITCH	*	di	
7215	SWITCH	*	di	
7216	SWITCH	*	di	
7218	SWITCH	*	di	
7219	RESTART	*	*	slspool: unable to allocate spooler argument buffer.
7223	RESTART	*	*	slspool: fork error, unable to spawn spool process. - Resource limit reached
7224	SWITCH	*	*	
7229	SWITCH	*	*	
7232	SWITCH	*	*	
7234	SWITCH	*	*	
7238	SWITCH	*	*	slemcl: close error.
7250	RESTART	*	*	
7251	RESTART	*	*	
7252	RESTART	*	*	
7253	RESTART	*	*	
7258	RESTART	*	*	
7259	RESTART	*	*	
7263	SWITCH	*	*	
7269	SWITCH	*	*	
7279	SWITCH	*	*	
7280	RESTART	*	*	
7296	SWITCH	*	*	
7297	SWITCH	*	*	
7306	RESTART	*	*	
7310	SWITCH	*	*	
7315	SWITCH	*	*	
7321	SWITCH	*	*	
7322	SWITCH	*	*	
7324	RESTART	*	*	
7325	RESTART	*	*	
7351	SWITCH	*	*	
7361	RESTART	*	*	
7404	SWITCH	*	*	
7414	RESTART	*	*	
7415	RESTART	*	*	

Error Number	Action	Connection State	New State	Message
7417	SWITCH	*	*	
7418	SWITCH	*	*	
7419	SWITCH	*	*	
7430	SWITCH	*	*	
7455	SWITCH	*	*	
7456	SWITCH	*	*	
7466	SWITCH	*	*	
7470	SWITCH	*	*	
7475	SWITCH	*	*	
7476	SWITCH	*	*	
7477	SWITCH	*	*	
7478	SWITCH	*	*	
7479	SWITCH	*	*	
7481	SWITCH	*	*	
9706	SWITCH	*	*	
9716	SWITCH	*	*	
9718	RESTART	*	*	
9740	SWITCH	*	*	
9748	SWITCH	*	*	
9747	RESTART	*	*	
9749	RESTART	*	*	
9751	RESTART	*	*	
9755	RESTART	*	*	
9757	RESTART	*	*	
9756	SWITCH	*	*	
9758	SWITCH	*	*	
9761	RESTART	*	*	
9765	RESTART	*	*	
9779	RESTART	*	*	
9829	RESTART	*	*	
9831	SWITCH	*	*	
9834	SWITCH	*	*	
9836	SWITCH	*	*	
9838	SWITCH	*	*	
9837	RESTART	*	*	
9844	RESTART	*	*	

Error Number	Action	Connection State	New State	Message
9845	RESTART	*	*	
9846	RESTART	*	*	
9847	RESTART	*	*	
9853	SWITCH	*	*	
9854	SWITCH	*	*	
9856	RESTART	*	*	
9874	SWITCH	*	*	
9876	SWITCH	*	*	
9877	RESTART	*	*	
9878	RESTART	*	*	
9879	RESTART	*	*	
9885	RESTART	*	*	
9888	RESTART	*	*	
9894	RESTART	*	*	
9909	RESTART	*	*	
9912	RESTART	*	*	
9913	RESTART	*	*	
9919	SWITCH	*	*	
9943	RESTART	*	*	
9947	RESTART	*	*	
9948	SWITCH	*	*	
9949	SWITCH	*	*	
9950	SWITCH	*	*	
12505	STOP	*	*	TNS:listener could not resolve SID given in connect descriptor.Check listener configuration file.
12541	STOP	*	*	TNS:no listener. Please verify connect_string property, listener and TNSconfiguration.
12545	SWITCH	*	*	Please check HA-Oracle parameters. Connect failed because target host or object does not exist
27100	STOP	*	*	Shared memory realm already exists
98765	STOP	*	*	Database role queried from database does not match the Oracle Solaris Cluster resource's dataguard role configuration.
99999	RESTART	*	di	Monitor detected death of Oracle background processes.

TABLE B-2 Preset Actions for Logged Alerts

Alert String	Action	Connection State	New State	Message
ORA-07265	SWITCH	*	di	Semaphore access problem
found dead multi-threaded server	NONE	*	*	Warning: Multi-threaded Oracle server process died (restarted automatically)
found dead dispatcher	NONE	*	*	Warning: Oracle dispatcher process died (restarted automatically)

♦ ♦ ♦ APPENDIX C

Sample Configurations for Oracle ASM with HA for Oracle Database

This appendix contains diagrams that explain various sample configurations of Oracle Automatic Storage Management (Oracle ASM) with HA for Oracle Database. The diagrams in this section provide information about the dependencies between HA for Oracle Database resources and Oracle ASM services.

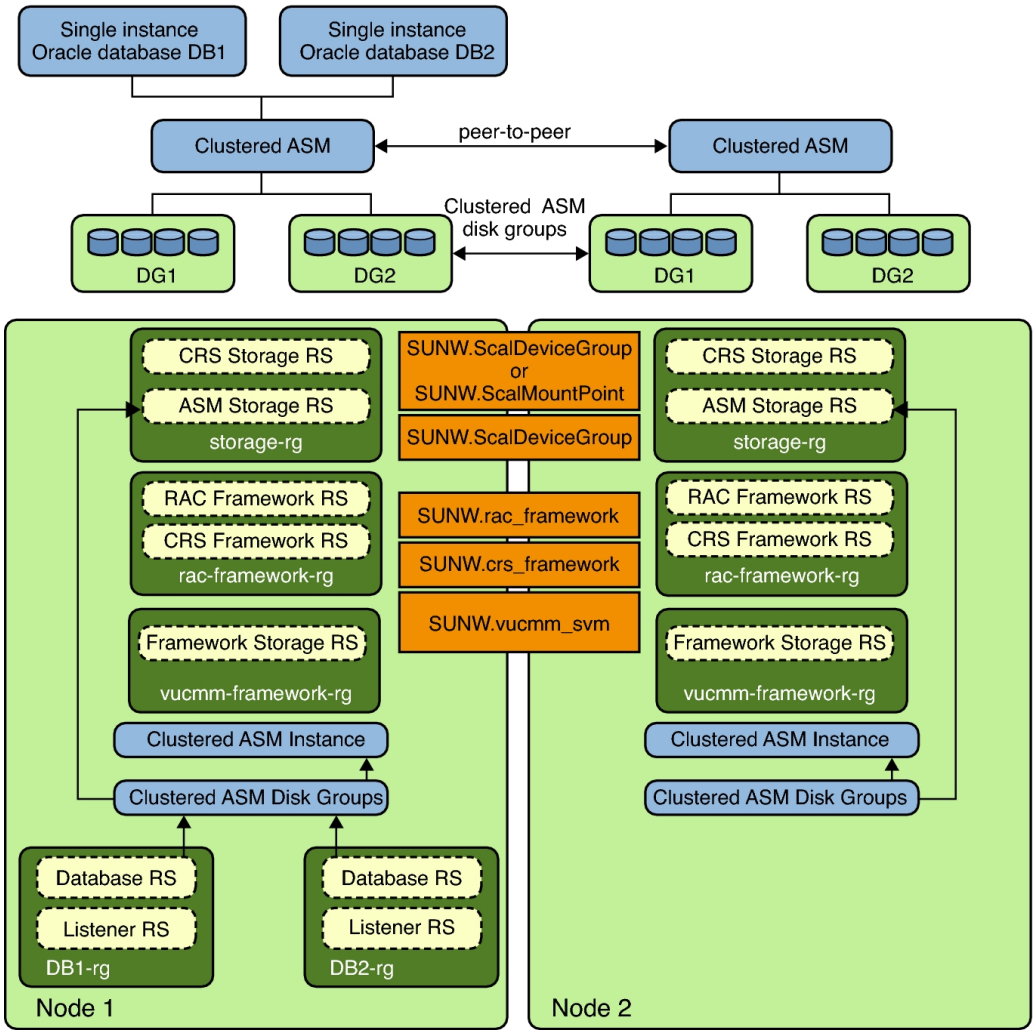
Clustered Oracle ASM with Clustered Disk Groups

This section contains sample configurations of single instance Oracle databases with clustered Oracle ASM instances and clustered Oracle ASM disk groups. There are two diagrams and the second diagram is a continuation of the first diagram.

The following diagram represents clustered Oracle ASM instances that serves two single instance Oracle databases, `DB1` and `DB2` on `Node1`. The databases `DB1` or `DB2` can use either one of the Oracle ASM disk groups `DG1` and `DG2` or both the disk groups as these are clustered Oracle ASM disk groups. The upper part of the diagram shows the relationships of the Oracle Database instances with the clustered Oracle ASM instances on `Node1` and `Node2`. The clustered Oracle ASM instances manage two Oracle ASM disk groups on both nodes at the same time. The lower part of the diagram represents the existing Oracle Solaris Cluster resource group and resources for single instance Oracle databases and their requirement for clustered Oracle ASM services.

The dotted box represents existing HA for Oracle Database resources with the new Oracle ASM resources. The arrows represent new dependencies between HA for Oracle Database and clustered Oracle ASM services.

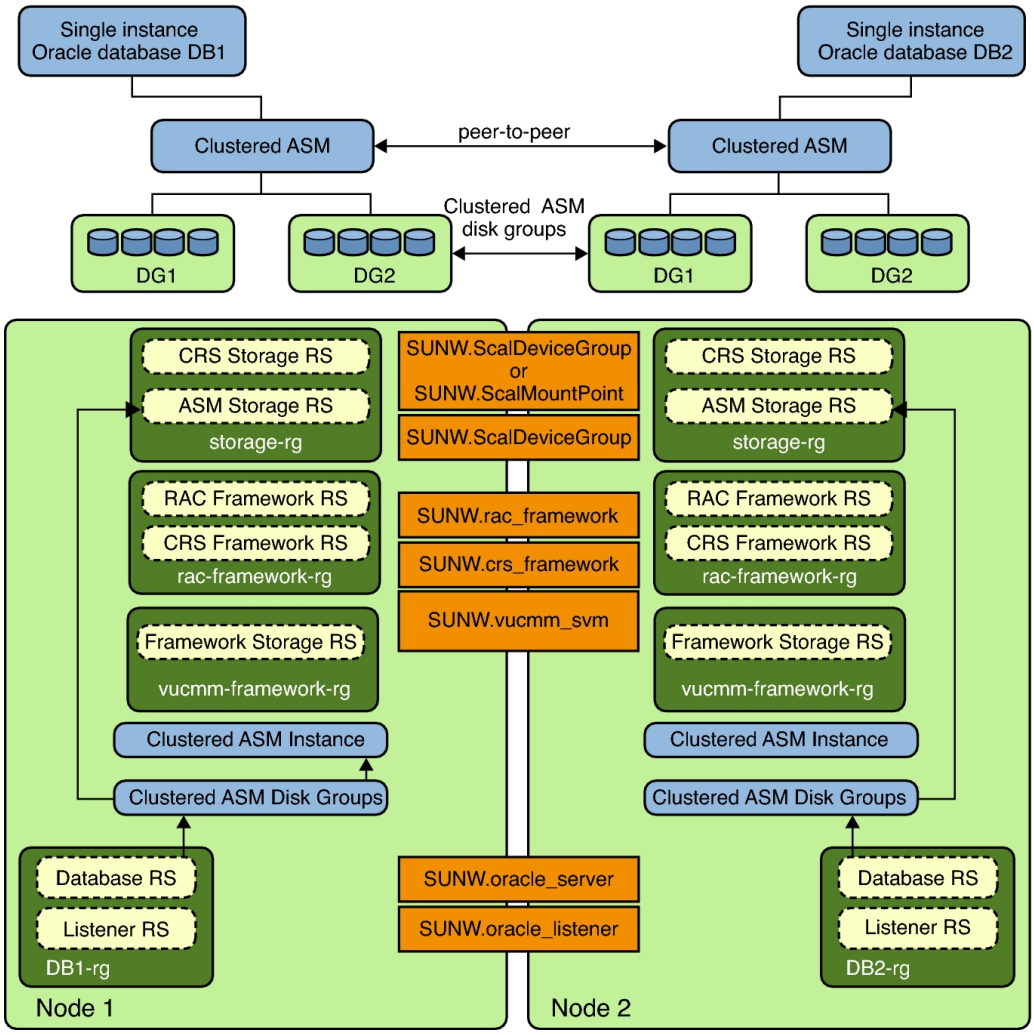
FIGURE C-1 Clustered Oracle ASM with Clustered Disk Groups [1]



In the following diagram, a single instance Oracle Database `DB2` that shares Oracle ASM disk groups `DG1` and `DG2` is now running on `Node2` after a failover of resource groups `DB2-rg` to `Node2`. The upper part of the diagram shows the relationships of the Oracle Database instances with the clustered Oracle ASM instances on `Node1` and `Node2`. The clustered Oracle ASM instances manage two Oracle ASM disk groups on both nodes at the same time. The lower part of the diagram represents the existing Oracle Solaris Cluster resource group and resources for single instance Oracle databases and their requirement for clustered Oracle ASM services. If the

storage type is hardware RAID, the resource types `SUNW.ScaleDeviceGroup` and `SUNW.rac_svm` are not required.

FIGURE C-2 Clustered Oracle ASM with Clustered Disk Groups [2]



Index

P

R

S

T

U

V

W

Z