Sun Cluster Data Service for Oracle Guide for Solaris OS

Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95054
U.S.A.

Part No: 819–0694–10
August 2005, Revision A

050506@11223

Contents

Tables

Preface

Sun Cluster Data Service for Oracle Guide for Solaris OS explains how to install and configure Sun™ Cluster HA for Oracle on both SPARC® and x86 based systems.

Note – In this document, the term "x86" refers to the Intel 32-bit family of microprocessor chips and compatible microprocessor chips made by AMD.

This document is intended for system administrators with extensive knowledge of Sun software and hardware. Do not use this document as a planning or presales guide. Before reading this document, you should have already determined your system requirements and purchased the appropriate equipment and software.

The instructions in this document assume knowledge of the Solaris™ Operating System and expertise with the volume manager software that is used with Sun Cluster.

Note – Sun Cluster software runs on two platforms, SPARC and x86. The information in this document pertains to both platforms unless otherwise specified in a special chapter, section, note, bulleted item, figure, table, or example.

UNIX Commands

This document contains information about commands that are specific to installing and configuring Sun Cluster data services. The document does *not* contain comprehensive information about basic UNIX® commands and procedures, such as shutting down the system, booting the system, and configuring devices. Information about basic UNIX commands and procedures is available from the following sources:

- Online documentation for the Solaris Operating System
- Solaris Operating System man pages
- Other software documentation that you received with your system

Typographic Conventions

The following table describes the typographic changes that are used in this book.

TABLE P–1 Typographic Conventions

Typeface or Symbol	Meaning	Example
`AaBbCc123`	The names of commands, files, and directories, and onscreen computer output	Edit your `.login` file. Use `ls -a` to list all files. `machine_name% you have mail.`
`AaBbCc123`	What you type, contrasted with onscreen computer output	`machine_name%` **`su`** `Password:`
AaBbCc123	Command-line placeholder: replace with a real name or value	The command to remove a file is `rm` *filename*.
AaBbCc123	Book titles, new terms, and terms to be emphasized	Read Chapter 6 in the *User's Guide*. Perform a *patch analysis*. Do *not* save the file. [Note that some emphasized items appear bold online.]

Shell Prompts in Command Examples

The following table shows the default system prompt and superuser prompt for the C shell, Bourne shell, and Korn shell.

TABLE P–2 Shell Prompts

Shell	Prompt
C shell prompt	`machine_name%`
C shell superuser prompt	`machine_name#`
Bourne shell and Korn shell prompt	`$`
Bourne shell and Korn shell superuser prompt	`#`

Related Documentation

Information about related Sun Cluster topics is available in the documentation that is listed in the following table. All Sun Cluster documentation is available at `http://docs.sun.com`.

Topic	Documentation
Data service administration	*Sun Cluster Data Services Planning and Administration Guide for Solaris OS*
	Individual data service guides
Concepts	*Sun Cluster Concepts Guide for Solaris OS*
Overview	*Sun Cluster Overview for Solaris OS*
Software installation	*Sun Cluster Software Installation Guide for Solaris OS*
System administration	*Sun Cluster System Administration Guide for Solaris OS*
Hardware administration	*Sun Cluster 3.0-3.1 Hardware Administration Manual for Solaris OS*
	Individual hardware administration guides
Data service development	*Sun Cluster Data Services Developer's Guide for Solaris OS*
Error messages	*Sun Cluster Error Messages Guide for Solaris OS*
Command and function reference	*Sun Cluster Reference Manual for Solaris OS*

For a complete list of Sun Cluster documentation, see the release notes for your release of Sun Cluster at `http://docs.sun.com`.

Related Third-Party Web Site References

Third-party URLs that are referenced in this document provide additional related information.

Note – Sun is not responsible for the availability of third-party web sites mentioned in this document. Sun does not endorse and is not responsible or liable for any content, advertising, products, or other materials that are available on or through such sites or resources. Sun will not be responsible or liable for any actual or alleged damage or loss caused or alleged to be caused by or in connection with use of or reliance on any such content, goods, or services that are available on or through such sites or resources.

Documentation, Support, and Training

Sun Function	URL	Description
Documentation	`http://www.sun.com/documentation/`	Download PDF and HTML documents, and order printed documents
Support and Training	`http://www.sun.com/supportraining/`	Obtain technical support, download patches, and learn about Sun courses

Product Training

Sun Microsystems offers training in many Sun technologies through a variety of instructor-led courses and self-paced courses. For information about the training courses that Sun offers and to enroll in a class, visit Sun Training at `http://training.sun.com/`.

Getting Help

If you have problems installing or using Sun Cluster, contact your service provider and provide the following information:

- Your name and email address (if available)
- Your company name, address, and phone number
- The model and serial numbers of your systems
- The release number of the Solaris Operating System (for example, Solaris 8)
- The release number of Sun Cluster (for example, Sun Cluster 3.0)

Use the following commands to gather information about each node on your system for your service provider.

Command	Function
`prtconf -v`	Displays the size of the system memory and reports information about peripheral devices
`psrinfo -v`	Displays information about processors
`showrev –p`	Reports which patches are installed
SPARC: `prtdiag -v`	Displays system diagnostic information
`scinstall -pv`	Displays Sun Cluster release and package version information

Also have available the contents of the `/var/adm/messages` file.

Installing and Configuring Sun Cluster HA for Oracle

This chapter explains how to install and configure Sun Cluster HA for Oracle.

This chapter contains the following sections.

Note – You can use SunPlex™ Manager to install and configure this data service. See the SunPlex Manager online help for details.

Overview of the Installation and Configuration Process for Sun Cluster HA for Oracle

The following table summarizes the tasks for installing and configuring Sun Cluster HA for Oracle. The table also provides cross-references to detailed instructions for performing the tasks. Perform these tasks in the order that they are listed.

TABLE 1 Task Map: Installing and Configuring HA for Oracle

Task	Cross-Reference
Plan the Sun Cluster HA for Oracle installation and configuration	"Planning the Sun Cluster HA for Oracle Installation and Configuration" on page 15
Prepare the nodes and disks	"Preparing the Nodes and Disks" on page 16
Install the Oracle software	"How to Install the Oracle Software" on page 19
Verify the Oracle installation	"How to Verify the Oracle Installation" on page 20
Create an Oracle database	"How to Create an Oracle Database" on page 21
Set up Oracle database permissions	"How to Set Up Oracle Database Permissions" on page 22
Install the Sun Cluster HA for Oracle packages	"Installing the Sun Cluster HA for Oracle Packages" on page 26
Register and configure Sun Cluster HA for Oracle	"How to Register and Configure Sun Cluster HA for Oracle" on page 29
Verify the Sun Cluster HA for Oracle installation	"Verifying the Sun Cluster HA for Oracle Installation" on page 34
Tune the Sun Cluster HA for Oracle fault monitor	"Tuning the Sun Cluster HA for Oracle Fault Monitors" on page 36
(Optional) Customize the Sun Cluster HA for Oracle server fault monitor	"Customizing the Sun Cluster HA for Oracle Server Fault Monitor" on page 40
(Optional) Upgrade Sun Cluster HA for Oracle resource types	"Upgrading Sun Cluster HA for Oracle Resource Types" on page 50

Planning the Sun Cluster HA for Oracle Installation and Configuration

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

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 Sun Cluster HA for Oracle. These requirements apply to Sun Cluster HA for Oracle only. You must meet these requirements before you proceed with your Sun Cluster HA for Oracle installation and configuration.

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

- **Oracle application files –** These files include Oracle 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 Sun Cluster Data Services" in *Sun Cluster Data Services Planning and Administration Guide for Solaris OS* for the advantages and disadvantages of placing the Oracle 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 the highly available local file system or the cluster file system as either raw devices or regular files.

Configuration Planning Questions

Use the questions in this section to plan the installation and configuration of Sun Cluster HA for Oracle. Write the answers to these questions in the space that is provided on the data service worksheets in "Configuration Worksheets" in *Sun Cluster Data Services Planning and Administration Guide for Solaris OS*.

- What resource groups will you use for network addresses and application resources and the dependencies between them?

- What is the logical hostname (for failover services) or shared address (for scalable services) for clients that will access the data service?
- Where will the system configuration files reside?

  See "Configuration Guidelines for Sun Cluster Data Services" in *Sun Cluster Data Services Planning and Administration Guide for Solaris OS* for the advantages and disadvantages of placing the Oracle binaries on the local file system rather than the cluster file system.

Preparing the Nodes and Disks

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

▼ How to Prepare the Nodes

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

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

Note – Consult the Oracle documentation before you perform this procedure.

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

Steps

1. **Become superuser on all of the cluster members.**
2. **Configure the cluster file system for Sun Cluster HA for Oracle.**

   If raw devices contain the databases, configure the global devices for raw device access. See the *Sun Cluster Software Installation Guide for Solaris OS* for information about how to configure global devices.

   If you use the Solstice DiskSuite™/Solaris Volume Manager software, configure the Oracle software to use UNIX file system (UFS) logging on mirrored metadevices or raw-mirrored metadevices. See the Solstice DiskSuite/Solaris Volume Manager documentation for more information about how to configure raw-mirrored metadevices.

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

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

---

4. **On each node, create an entry for the database administrator (DBA) group in the `/etc/group` file, and add potential users to the group.**

   You typically name the DBA group *dba*. Verify that the `root` and *oracle* users are members of the *dba* group, and add entries as necessary for other DBA users. Ensure that the group IDs are the same on all of the nodes that run Sun Cluster HA for Oracle, as the following example illustrates.

   dba`:*:520:root,oracle`

   You can create group entries in a network name service (for example, NIS or NIS+). If you create group entries in this way, add your entries to the local `/etc/inet/hosts` file to eliminate dependency on the network name service.

5. **On each node, create an entry for the Oracle user ID (*oracle*).**

   You typically name the Oracle user ID *oracle*. The following command updates the `/etc/passwd` and `/etc/shadow` files with an entry for the Oracle user ID.

   `# useradd -u 120 -g` *dba* `-d /`*Oracle-home oracle*

   Ensure that the *oracle* user entry is the same on all of the nodes that run Sun Cluster HA for Oracle.

▼ How to Configure Oracle Database Access With Solstice DiskSuite

Use this procedure to configure the Oracle database with Solstice DiskSuite volume manager.

Steps

1. **Configure the disk devices for the Solstice DiskSuite software to use.**

   See the *Sun Cluster Software Installation Guide for Solaris OS* for information about how to configure the Solstice DiskSuite 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 node that can master the Oracle resource group.**

      `# chown` *oracle* `/dev/md/`metaset`/rdsk/d`*n*
      `# chgrp` *dba* `/dev/md/`*metaset*`/rdsk/d`*n*

```
# chmod 600 /dev/md/metaset/rdsk/dn
```

metaset	Specifies the name of the diskset
/rdsk/d*n*	Specifies the name of the raw disk device within the *metaset* diskset

b. Verify that the changes are effective.

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

▼ SPARC: How to Configure Oracle Database Access With VERITAS Volume Manager

Use this procedure to configure the Oracle database with VERITAS Volume Manager software.

Steps

1. Configure the disk devices for the VxVM software to use.

See the *Sun Cluster Software Installation Guide for Solaris OS* for information about how to configure VERITAS Volume Manager.

2. If you use raw devices to contain the databases, run the following commands on the current disk-group primary to change each device'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 command for each raw device.

```
# vxedit -g diskgroup set user=oracle group=dba mode=600 volume
```

diskgroup	Specifies the name of the disk group
volume	Specifies the name of the raw volume within the disk group

b. Verify that the changes are effective.

```
# ls -lL /dev/vx/rdsk/diskgroup/volume
```

c. Reregister the disk device group with the cluster to keep the VxVM namespace consistent throughout the cluster.

```
# scconf -c -D name=diskgroup
```

Installing the Oracle Software

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

▼ How to Install the Oracle Software

Steps

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

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

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

   For more information about installation locations, see "Preparing the Nodes and Disks" on page 16.

3. **Install the Oracle software.**

   Regardless of where you install the Oracle software, modify each node's `/etc/system` files as you would in standard Oracle installation procedures. Then reboot.

   Log in as *oracle* to ensure ownership of the entire directory before you perform this step. See the appropriate Oracle installation and configuration guides for instructions about how to install Oracle software.

4. **(Optional) If you are using Sun Cluster HA for Oracle with Oracle 10g, prevent the Oracle `cssd` daemon from being started.**

   Remove the entry for the Oracle `cssd` daemon from the `/etc/inittab` file on the node where the Oracle software is installed. To remove this entry, remove the following line from the `/etc/inittab` file:

   ```
   h1:23:respawn:/etc/init.d/init.cssd run >/dev/null 2>&1 > </dev/null
   ```

   Sun Cluster HA for Oracle does not require the Oracle `cssd` daemon. Therefore, removal of this entry does *not* affect the operation of Oracle 10g with Sun Cluster HA for Oracle. If your Oracle installation changes so that the Oracle `cssd` daemon is required, restore the entry for this daemon to the `/etc/inittab` file.

Caution – If you are using Oracle 10g Real Application Clusters, do *not* remove the entry for the `cssd` daemon from the `/etc/inittab` file.

If you remove the entry for the Oracle `cssd` daemon from the `/etc/inittab` file, you prevent unnecessary error messages from being displayed. Otherwise, an attempt by the `init`(1M) command to start the Oracle `cssd` daemon might cause such error messages to be displayed. These error messages are displayed if the Oracle binary files are installed on a highly available local file system or on the cluster file system. The messages are displayed repeatedly until the file system where the Oracle binary files are installed is mounted.

These error messages are as follows:

```
INIT: Command is respawning too rapidly. Check for possible errors.
id:  h1 "/etc/init.d/init.cssd run >/dev/null 2>&1 >/dev/null"

Waiting for filesystem containing $CRSCTL.
```

If you are using Sun Cluster HA for Oracle on the x86 platform, unnecessary error messages about the unavailability of the UNIX Distributed Lock Manager (Oracle UDLM) might also be displayed.

These messages are displayed if the following events occur:

- A node is running in noncluster mode. In this situation, file systems that Sun Cluster controls are never mounted.
- A node is booting. In this situation, the messages are displayed repeatedly until Sun Cluster mounts the file system where the Oracle binary files are installed.
- Oracle is started on or fails over to a node where the Oracle installation was *not* originally run. In such a configuration, the Oracle binary files are installed on a highly available local file system. In this situation, the messages are displayed on the console of the node where the Oracle installation was run.

Verifying the Oracle Installation and Configuration

This section contains the procedure that you need to verify the Oracle installation and configuration.

▼ How to Verify the Oracle Installation

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

Steps

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 21.

Creating an Oracle Database

This section contains the procedure to configure and create the initial Oracle database in a Sun Cluster environment. If you create and configure additional databases, omit the procedure "How to Create an Oracle Database" on page 21.

▼ How to Create an Oracle Database

Steps

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 Nodes and Disks" on page 16 for information about installation locations.

   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 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 one utility from the following list:**

   - The Oracle installer
   - The Oracle `sqlplus`(1M) command

During creation, ensure that all of the database-related files are placed in the appropriate location, either on shared global devices or on the cluster 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 Sun Cluster HA for Oracle fault monitor uses this view.

Next Steps When you have completed the work in this section, go to "Setting Up Oracle Database Permissions" on page 22.

Setting Up Oracle Database Permissions

Perform the procedure in this section to set up Oracle database permissions.

▼ How to Set Up Oracle Database Permissions

Steps

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

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

     ```
     # sqlplus  "/ as sysdba"

     sql>    grant connect, resource to 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 create session to user;
     sql>    grant create table to user;

     sql>    exit;
     #
     ```

     You can use this method for all of the supported Oracle releases.

   - **To use the 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 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 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$`.
- *user* is the user for whom you are enabling Solaris authentication. Ensure that this user owns the files under the `$ORACLE_HOME` directory.

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

---

2. Configure NET8 for the Sun Cluster software.

The `listener.ora` file must be accessible from all of the nodes that are in the cluster. Place these files either under the cluster file system or in the local file system of each node that can potentially run the Oracle 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 variable in a user-environment file. For information about Oracle variables, see the Oracle documentation. You must also run the `scrgadm`(1M) command to set the resource extension parameter `User_env`, which sources the user-environment file. See "`SUNW.oracle_listener` Extension Properties" on page 59 or "`SUNW.oracle_server` Extension Properties" on page 55 for format details.

Sun Cluster HA for Oracle imposes no restrictions on the listener name—it can be any valid Oracle listener name.

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

```
LISTENER =
    (ADDRESS_LIST =
            (ADDRESS =
                (PROTOCOL = TCP)
                    (HOST = logical-hostname) <- use logical hostname
                (PORT = 1527)
            )
    )
.
.
SID_LIST_LISTENER =
    .
            .
                        (SID_NAME = SID) <- Database name,
default is ORCL
```

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
```

The following example shows how to update the `listener.ora` and `tnsnames.ora` files for the following Oracle instances.

Instance	Logical Host	Listener
ora8	hadbms3	LISTENER-ora8
ora9	hadbms4	LISTENER-ora9

The corresponding `listener.ora` entries are the following entries.

```
LISTENER-ora9 =
    (ADDRESS_LIST =
            (ADDRESS =
                (PROTOCOL = TCP)
                (HOST = hadbms4)
                (PORT = 1530)
            )
        )
SID_LIST_LISTENER-ora9 =
    (SID_LIST =
            (SID_DESC =
                (SID_NAME = ora9)
            )
        )
LISTENER-ora8 =
  (ADDRESS_LIST =
    (ADDRESS= (PROTOCOL=TCP) (HOST=hadbms3)(PORT=1806))
  )
SID_LIST_LISTENER-ora8 =
  (SID_LIST =
     (SID_DESC =
            (SID_NAME = ora8)
         )
  )
```

The corresponding `tnsnames.ora` entries are the following entries.

```
ora8 =
(DESCRIPTION =
   (ADDRESS_LIST =
            (ADDRESS = (PROTOCOL = TCP)
            (HOST = hadbms3)
            (PORT = 1806))
       )
    (CONNECT_DATA = (SID = ora8))
)
ora9 =
(DESCRIPTION =
  (ADDRESS_LIST =
        (ADDRESS =
                (PROTOCOL = TCP)
                (HOST = hadbms4)
                (PORT = 1530))
  )
      (CONNECT_DATA = (SID = ora9))
)
```

3. **Verify that the Sun Cluster software is installed and running on all of the nodes.**

```
# scstat
```

Next Steps Go to "Installing the Sun Cluster HA for Oracle Packages" on page 26 to install the Sun Cluster HA for Oracle packages.

Installing the Sun Cluster HA for Oracle Packages

If you did not install the Sun Cluster HA for Oracle packages during your initial Sun Cluster installation, perform this procedure to install the packages. Perform this procedure on each cluster node where you are installing the Sun Cluster HA for Oracle packages. To complete this procedure, you need the Sun Cluster Agents CD-ROM.

If you are installing more than one data service simultaneously, perform the procedure in "Installing the Software" in *Sun Cluster Software Installation Guide for Solaris OS*.

Install the Sun Cluster HA for Oracle packages by using one of the following installation tools:

- The Web Start program
- The `scinstall` utility

Note – If you are using Solaris 10, install these packages *only* in the global zone. To ensure that these packages are not propagated to any local zones that are created after you install the packages, use the `scinstall` utility to install these packages. Do *not* use the Web Start program.

▼ How to Install the Sun Cluster HA for Oracle Packages by Using the Web Start Program

You can run the Web Start program with a command-line interface (CLI) or with a graphical user interface (GUI). The content and sequence of instructions in the CLI and the GUI are similar. For more information about the Web Start program, see the `installer`(1M) man page.

Steps 1. **On the cluster node where you are installing the Sun Cluster HA for Oracle packages, become superuser.**

2. **(Optional) If you intend to run the Web Start program with a GUI, ensure that your `DISPLAY` environment variable is set.**

3. **Insert the Sun Cluster Agents CD-ROM into the CD-ROM drive.**

   If the Volume Management daemon `vold`(1M) is running and configured to manage CD-ROM devices, it automatically mounts the CD-ROM on the `/cdrom/cdrom0` directory.

4. **Change to the Sun Cluster HA for Oracle component directory of the CD-ROM.**

   The Web Start program for the Sun Cluster HA for Oracle data service resides in this directory.

   ```
   # cd /cdrom/cdrom0/components/SunCluster_HA_Oracle_3.1
   ```

5. **Start the Web Start program.**

   ```
   # ./installer
   ```

6. **When you are prompted, select the type of installation.**

   - To install only the C locale, select Typical.
   - To install other locales, select Custom.

7. **Follow the instructions on the screen to install the Sun Cluster HA for Oracle packages on the node.**

   After the installation is finished, the Web Start program provides an installation summary. This summary enables you to view logs that the Web Start program created during the installation. These logs are located in the `/var/sadm/install/logs` directory.

8. **Exit the Web Start program.**

9. **Remove the Sun Cluster Agents CD-ROM from the CD-ROM drive.**

   a. **To ensure that the CD-ROM is not being used, change to a directory that does *not* reside on the CD-ROM.**

   b. **Eject the CD-ROM.**

      ```
      # eject cdrom
      ```

Next Steps See "Registering and Configuring Sun Cluster HA for Oracle" on page 28 to register Sun Cluster HA for Oracle and to configure the cluster for the data service.

▼ How to Install the Sun Cluster HA for Oracle Packages by Using the `scinstall` Utility

Steps

1. **Load the Sun Cluster Agents CD-ROM into the CD-ROM drive.**
2. **Run the `scinstall` utility with no options.**
   This step starts the `scinstall` utility in interactive mode.
3. **Choose the menu option, Add Support for New Data Service to This Cluster Node.**
   The `scinstall` utility prompts you for additional information.
4. **Provide the path to the Sun Cluster Agents CD-ROM.**
   The utility refers to the CD as the "data services cd."
5. **Specify the data service to install.**
   The `scinstall` utility lists the data service that you selected and asks you to confirm your choice.
6. **Exit the `scinstall` utility.**
7. **Unload the CD from the drive.**

Next Steps See "Registering and Configuring Sun Cluster HA for Oracle" on page 28 to register Sun Cluster HA for Oracle and to configure the cluster for the data service.

Registering and Configuring Sun Cluster HA for Oracle

This section contains the procedures that you need to configure Sun Cluster HA for Oracle.

Setting Sun Cluster HA for Oracle Extension Properties

Use the extension properties in Appendix A to create your resources. Use the command `scrgadm -x parameter=value` to configure extension properties when you create your resource. Use the procedure in Chapter 2, "Administering Data Service Resources," in *Sun Cluster Data Services Planning and Administration Guide for*

Solaris OS 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 Appendix A, "Standard Properties," in *Sun Cluster Data Services Planning and Administration Guide for Solaris OS* for details about all Sun Cluster properties.

"SUNW.oracle_server Extension Properties" on page 55 describes the extension properties that you can set for the Oracle server. For the Oracle server, you are required to set only the following extension properties:

- ORACLE_HOME
- ORACLE_SID
- Alert_log_file
- Connect_string

▼ How to Register and Configure Sun Cluster HA for Oracle

Use this procedure to configure Sun Cluster HA for Oracle as a failover data service. This procedure assumes that you installed the data service packages during your initial Sun Cluster installation. If you did not install the Sun Cluster HA for Oracle packages as part of your initial Sun Cluster installation, go to "Installing the Sun Cluster HA for Oracle Packages" on page 26 to install the data service packages. Otherwise, use this procedure to configure the Sun Cluster HA for Oracle.

You must have the following information to perform this procedure.

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

Steps

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

2. **Run the `scrgadm` command to register the resource types for the data service.**

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

   ```
   # scrgadm -a -t SUNW.oracle_server
   # scrgadm -a -t SUNW.oracle_listener
   ```

`-a`	Adds the data service resource type.
`-t SUNW.oracle_`*type*	Specifies the predefined resource type name for your data service.

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

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

   ```
   # scrgadm -a -g resource-group [-h nodelist]
   ```

`-g` *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.
`-h` *nodelist*	Specifies an optional comma-separated list of physical node names or IDs that identify potential masters. The order here determines the order in which the nodes are considered as primary during failover.

   Note – Use the `-h` option to specify the order of the node list. If all of the nodes that are in the cluster are potential masters, you do not need to use the `-h` option.

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 Sun 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 network resource to the failover resource group.**

   ```
   # scrgadm -a -L -g resource-group -l logical-hostname [-n netiflist]
   ```

`-l` *logical-hostname*	Specifies a network resource. The network resource is the logical hostname or shared address (IP address) that clients use to access Sun Cluster HA for Oracle.
[`-n` *netiflist*]	Specifies an optional, comma-separated list that identifies the IP Networking Multipathing groups that are on each node. Each element in *netiflist* must be in the form of `netif@node`. `netif` can be given as an IP Networking Multipathing group name, such as `sc_ipmp0`. The node can be identified by the node name or node ID, such as `sc_ipmp0@1` or `sc_ipmp@phys-schost-1`.

Note – Sun Cluster does not currently support the use of the adapter name for `netif`.

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

```
# scrgadm -a -t SUNW.HAStoragePlus
```

7. **Create the resource `oracle-hastp-rs` of type `SUNW.HAStoragePlus`.**

```
# scrgadm -a -j oracle-hastp-rs -g oracle-rg -t SUNW.HAStoragePlus \

[If your database is on a raw device, specify the global device path.]
-x GlobalDevicePaths=ora-set1,/dev/global/dsk/d1 \

[If your database is on a Cluster File Service, specify
the global file system and local file system mount points.]
-x FilesystemMountPoints=/global/ora-inst,/global/ora-data/logs,/
local/ora-data \

[Set AffinityOn to true.]
-x AffinityOn=TRUE
```

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

8. **Run the `scswitch` command to complete the following tasks and bring the resource group oracle-rg online on a cluster node.**

Caution – Be sure to switch only at the resource group level. Switching at the device group level confuses the resource group, causing it to fail over.

- Move the resource group into a `MANAGED` state.
- Bring the resource group online.

This node is made the primary for device group `ora-set1` and raw device `/dev/global/dsk/d1`. Device groups that are associated with file systems such as `/global/ora-inst` and `/global/ora-data/logs` are also made primaries on this node.

```
# scswitch -Z -g oracle-rg
```

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

- Oracle server resource:

```
# scrgadm -a -j resource -g resource-group \
-t SUNW.oracle_server \
-x Connect_string=user/passwd \
-x ORACLE_SID=instance \
-x ORACLE_HOME=Oracle-home \
-x Alert_log_file=path-to-log \
-x Restart_type=entity-to-restart
-y resource_dependencies=storageplus-resource
```

- Oracle listener resource:

```
# scrgadm -a -j resource -g resource-group \
-t SUNW.oracle_listener \
-x LISTENER_NAME=listener \
-x ORACLE_HOME=Oracle-home
-y resource_dependencies=storageplus-resource
```

`-j` *resource*
Specifies the name of the resource to add.

`-g` *resource-group*
Specifies the name of the resource group into which the resources are to be placed.

`-t SUNW.oracle_server/listener`
Specifies the type of the resource to add.

`-x Alert_log_file=`*path-to-log*
Sets the path under `$ORACLE_HOME` for the server message log.

`-x 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 22. If you use Solaris authorization, type a slash (`/`) instead of the user name and password.

`-x ORACLE_SID=`*instance*
Sets the Oracle system identifier.

`-x LISTENER_NAME=`*listener*
Sets the name of the Oracle listener instance. This name must match the corresponding entry in `listener.ora`.

`-x ORACLE_HOME=`*Oracle-home*
Sets the path to the Oracle home directory.

`-x 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 server resource is restarted.

Note – Optionally, you can set additional extension properties that belong to the Oracle data service to override their default values. See "Setting Sun Cluster HA for Oracle Extension Properties" on page 28 for a list of extension properties.

---

10. Enable the resource and fault monitoring.

```
# scswitch -Z -g resource-group
```

`-Z`	Enables the resource and monitor, moves the resource group to the `MANAGED` state, and brings it online.
`-g` *resource-group*	Specifies the name of the resource group.

Example 1 Registering Sun Cluster HA for Oracle

The following example shows how to register Sun Cluster HA for Oracle on a two-node cluster.

Cluster Information
Node names: phys-schost-1, phys-schost-2
Logical Hostname: schost-1
Resource group: resource-group-1 (failover resource group)
Oracle Resources: oracle-server-1, oracle-listener-1
Oracle Instances: ora-lsnr (listener), ora-srvr (server)

```
(Add the failover resource group to contain all of the resources.)
# scrgadm -a -g resource-group-1

(Add the logical hostname resource to the resource group.)
# scrgadm -a -L -g resource-group-1 -l schost-1

(Register the Oracle resource types.)
# scrgadm -a -t SUNW.oracle_server
# scrgadm -a -t SUNW.oracle_listener

(Add the Oracle application resources to the resource group.)
# scrgadm -a -j oracle-server-1 -g resource-group-1 \
-t SUNW.oracle_server -x ORACLE_HOME=/global/oracle \
-x Alert_log_file=/global/oracle/message-log \
-x ORACLE_SID=ora-srvr -x Connect_string=scott/tiger

# scrgadm -a -j oracle-listener-1 -g resource-group-1 \
-t SUNW.oracle_listener -x ORACLE_HOME=/global/oracle \
-x LISTENER_NAME=ora-lsnr
```

(Bring the resource group online.)

```
# scswitch -Z -g resource-group-1
```

Where to Go From Here

Go to "Verifying the Sun Cluster HA for Oracle Installation" on page 34 after you register and configure Sun Cluster HA for Oracle.

Verifying the Sun Cluster HA for Oracle Installation

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

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

▼ How to Verify the Sun Cluster HA for Oracle Installation

Steps

1. **Log in as *oracle* to the node that currently masters the Oracle resource group.**
2. **Set the environment variables `ORACLE_SID` and `ORACLE_HOME`.**
3. **Confirm that you can start the Oracle instance from this node.**
4. **Confirm that you can connect to the Oracle instance.**

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

   ```
   # sqlplus user/passwd@tns_service
   ```

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

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

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

   The following example shows how to complete this step.

   ```
   # scswitch -z -g resource-group -h node
   ```

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

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

Oracle 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 nodes during failover. The physical hostname is a 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. The network resource is a logical hostname or a shared address. See "How to Set Up Oracle Database Permissions" on page 22.

Note – Oracle client-server connections cannot survive a Sun Cluster HA for Oracle switchover. The client application must be prepared to handle disconnection and reconnection or recovery as appropriate. A transaction monitor might simplify the application. Further, Sun Cluster HA for Oracle node recovery time is application dependent.

Location of Sun Cluster HA for Oracle Log Files

Each instance of the Sun Cluster HA for Oracle 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 server.
- The `/var/opt/SUNWscor/oracle_listener` directory contains log files for the Oracle listener.

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

Tuning the Sun Cluster HA for Oracle Fault Monitors

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

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

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

TABLE 2 Resource Types for Sun Cluster HA for Oracle Fault Monitors

Fault Monitor	Resource Type
Oracle server	`SUNW.oracle_server`
Oracle 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 Sun Cluster installations. Therefore, you should tune the Sun Cluster HA for Oracle fault monitors *only* if you need to modify this preset behavior.

Tuning the Sun Cluster HA for Oracle 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 Sun Cluster Data Services" in *Sun Cluster Data Services Planning and Administration Guide for Solaris OS*. Information about the Sun Cluster HA for Oracle fault monitors that you need to perform these tasks is provided in the subsections that follow.

Tune the Sun Cluster HA for Oracle fault monitors when you register and configure Sun Cluster HA for Oracle. For more information, see "Registering and Configuring Sun Cluster HA for Oracle" on page 28.

Operation of the Oracle Server Fault Monitor

The fault monitor for the Oracle 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

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
2. If the partition is healthy, determining whether the database is operational

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.

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 user. The ID of this user is specified during the preparation of the nodes as explained in "How to Prepare the Nodes" on page 16.

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 Sun Cluster HA for Oracle Server Fault Monitor" on page 40.

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 node.

- **Switch over.** The server fault monitor switches over the database server resource group to another 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

The Oracle software 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 2. To change the action that the server fault monitor performs, customize the server fault monitor as explained in "Customizing the Sun Cluster HA for Oracle Server Fault Monitor" on page 40.

Operation of the Oracle Listener Fault Monitor

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

If the listener is running, the Oracle 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 listener security is enabled, a probe by the listener fault monitor might return Oracle 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 in 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.

Customizing the Sun Cluster HA for Oracle Server Fault Monitor

Customizing the Sun Cluster HA for Oracle 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 Sun Cluster HA for Oracle 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 Sun Cluster HA for Oracle 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 Sun Cluster HA for Oracle 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 Sun Cluster HA for Oracle 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 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.

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 Sun Cluster HA for Oracle 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

ERROR_TYPE	Data Type	Meaning
SCAN_LOG	Quoted regular expression	A string in an error message that Oracle has logged to the Oracle 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 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 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.

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 error 4031: unable to allocate *num-bytes* bytes of shared memory. However, this Oracle 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 2 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 error 4030: `out of process memory when trying to allocate` *num-bytes* `bytes` is restart. This Oracle 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 3 Ignoring a DBMS Error

```
{
ERROR_TYPE=DBMS_ERROR;
ERROR=4030;
```

EXAMPLE 3 Ignoring a DBMS Error *(Continued)*

```
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

The Oracle 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 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 has logged to the Oracle 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 4 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 5 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=*;
```

EXAMPLE 5 Changing the Maximum Number of Consecutive Timed-Out Probes *(Continued)*

```
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 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`(1) command or the `rdist`(1) 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

Steps

1. **On a cluster node, become superuser.**

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.

   ```
   # scrgadm -c -j server-resource\
     -x custom_action_file=filepath
   ```

   `-j` *server-resource*
   Specifies the `SUNW.oracle_server` resource

   `-x custom_action_file=`*filepath*
   Specifies the absolute path of the custom action file

Upgrading Sun Cluster HA for Oracle Resource Types

The resource types for the Sun Cluster HA for Oracle data service are as follows:

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

Upgrade these resource types if all conditions in the following list apply:

- You are upgrading from an earlier version of the Sun Cluster HA for Oracle data service.
- You need to use the new features of this data service.

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

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 subsections that follow.

Information for Registering the New Resource Type Version

The relationship between the version of the `SUNW.oracle_listener` resource type and the release of Sun Cluster data services is shown in the following table. The release of Sun Cluster data services indicates the release in which the version of the resource type was introduced. The table also summarizes the changes that were introduced in each new version.

`SUNW.oracle_listener` Resource Type Version	Sun Cluster Data Services Release	Summary of Changes
1	1.0	Not applicable
3.1	3.1 5/03	Directives to enable upgrades of this resource type introduced

SUNW.oracle_listener Resource Type Version	Sun Cluster Data Services Release	Summary of Changes
4	3.1 4/04	Probe_timeout extension property introduced
5	3.1 8/05	Default value of the Probe_timeout extension property increased to 180 seconds

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

- `scrgadm -p`
- `scrgadm -pv`

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

Information for Migrating Existing Instances of the 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 use the features of the `SUNW.oracle_listener` resource type that were introduced in version 3.1 4/04, the required value of the `Type_version` property is `4`.
- If you need to use the features of the `SUNW.oracle_listener` resource type that were introduced in version 3.1 8/05, the required value of the `Type_version` property is `5`.
- If you need to specify the timeout value in seconds that the fault monitor uses to probe an Oracle listener, set the `Probe_timeout` extension property. For more information, see "`SUNW.oracle_listener` Extension Properties" on page 59.

Note – If you are using version 4 of the `SUNW.oracle_listener` resource type, upgrade to version 4 *only* if you require the new default values. If the default values in version 4 are satisfactory, you do not need to upgrade.

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

EXAMPLE 6 Editing an Instance of the `SUNW.oracle_listener` Resource Type

```
# scrgadm -cj oracle-lrs -y Type_version=4 \
  -x probe_timeout=60
```

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

EXAMPLE 6 Editing an Instance of the `SUNW.oracle_listener` Resource Type *(Continued)*

- The `SUNW.oracle_listener` resource is named `oracle-lrs`.
- The `Type_version` property of this resource is set to `4`.
- The timeout value in seconds that the fault monitor uses to probe an Oracle 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 subsections that follow.

Information for Registering the New Resource Type Version

The relationship between the version of the `SUNW.oracle_server` resource type and the release of Sun Cluster data services is shown in the following table. The release of Sun Cluster data services indicates the release in which the version of the resource type was introduced. The table also summarizes the changes that were introduced in each new version.

`SUNW.oracle_server` Resource Type Version	Sun Cluster Data Services Release	Summary of Changes
1	1.0	Not applicable
1.1	3.0 5/02 asynchronous release	Directives to enable upgrades of this resource type introduced `Auto_End_Bkp` extension property introduced
3.1	3.1 5/03	`Restart_type` extension property introduced
4	3.1 10/03	`Custom_action_file` extension property introduced
5	3.1 8/05	Default values of extension properties changed as follows: ■ Default value of `Probe_timeout` extension property increased to 300 seconds ■ Default value of `Restart_type` extension property changed to `RESOURCE_RESTART`

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

- `scrgadm -p`
- `scrgadm -pv`

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

Information for Migrating Existing Instances of the 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 need to use the features of the `SUNW.oracle_server` resource type that were introduced in version 3.1 10/03, the required value of the `Type_version` property is `4`.
- If you need to use the features of the `SUNW.oracle_server` resource type that were introduced in version 3.1 8/05, the required value of the `Type_version` property is `5`.
- If you customized the behavior of the server fault monitor, set the `Custom_action_file` extension property. For more information, see "Customizing the Sun Cluster HA for Oracle Server Fault Monitor" on page 40.

Note – If you are using version 4 of the `SUNW.oracle_server` resource type, upgrade to version 4 *only* if you require the new default values. If the default values in version 4 are satisfactory, you do not need to upgrade.

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

EXAMPLE 7 Editing an Instance of the `SUNW.oracle_server` Resource Type

```
# scrgadm -cj oracle-srs -y Type_version=4 \
  -x custom_action_file=/opt/SUNWscor/oracle_server/etc/srv_mon_cust_actions
```

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 `4`.
- Custom behavior for the fault monitor of this resource is specified in the file `/opt/SUNWscor/oracle_server/etc/srv_mon_cust_actions`.

APPENDIX **A**

Sun Cluster HA for Oracle Extension Properties

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

- "`SUNW.oracle_server` Extension Properties" on page 55
- "`SUNW.oracle_listener` Extension Properties" on page 59

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

`Alert_log_file` (string)
Specifies the absolute path of the Oracle alert log file. The Oracle software logs alerts in this file. The Oracle 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 Appendix B. To change the action that the server fault monitor performs, customize the server fault monitor as explained in "Customizing the Sun Cluster HA for Oracle Server Fault Monitor" on page 40.

Default: None

Range: Minimum = 1

Tunable: Any time

`Auto_End_Bkp` (Boolean)
Specifies whether the following recovery actions are performed if an Oracle 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 Sun Cluster HA for Oracle 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

`Connect_string` (string)
The Oracle database user ID and password that the server fault monitor uses to connect to the database.

Specify this property as follows:

userid/password

As part of the setup of Sun Cluster HA for Oracle, you must define the database user ID and password before enabling the server resource and its fault monitor. To use Solaris authentication, type a slash (/) instead of a user ID and password.

Default: None

Range: Minimum = 1

Tunable: Any time

`Custom_action_file` (string)
The absolute path of the file that defines the custom behavior of the Sun Cluster HA for Oracle server fault monitor.

Default: ""

Range: None

Tunable: Any time

Introduced in release: 3.1 10/03

`Debug_level` (integer)
The level to which debug messages from the Oracle 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 server component.

Default: 1, which logs `syslog` messages

Range: 0– 100

Tunable: Any time

`ORACLE_HOME` (string)
The path to the Oracle home directory.

Default: None

Range: Minimum = 1

Tunable: When disabled

`ORACLE_SID` (string)
The Oracle system identifier.

Default: None

Range: Minimum = 1

Tunable: When disabled

`Parameter_file` (string)
The Oracle parameter file. If the Oracle parameter file is not specified, this property defaults to Oracle's 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 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

`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 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

`Debug_level` (integer)
The level to which debug messages from the Oracle listener component are logged. When the debug level is increased, more debug messages are written to the log files.

Default: 1, which logs `syslog` messages

Range: 0 – 100

Tunable: Any time

`LISTENER_NAME` (string)
The name of the Oracle 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 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 listener.

Default: 180

Range: 1– 99,999

Tunable: Any time

Introduced in release: 3.1 4/04

`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 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 Database Management System (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 1.
- Logged alerts for which an action is preset are listed in Table 2.

TABLE 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

TABLE 1 Preset Actions for DBMS Errors *(Continued)*

Error Number	Action	Connection State	New State	Message
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
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

TABLE 1 Preset Actions for DBMS Errors *(Continued)*

Error Number	Action	Connection State	New State	Message
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
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	*	*	

TABLE 1 Preset Actions for DBMS Errors *(Continued)*

Error Number	Action	Connection State	New State	Message
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	
7214	SWITCH	*	di	

TABLE 1 Preset Actions for DBMS Errors *(Continued)*

Error Number	Action	Connection State	New State	Message
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	*	*	

TABLE 1 Preset Actions for DBMS Errors *(Continued)*

Error Number	Action	Connection State	New State	Message
7325	RESTART	*	*	
7351	SWITCH	*	*	
7361	RESTART	*	*	
7404	SWITCH	*	*	
7414	RESTART	*	*	
7415	RESTART	*	*	
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	*	*	

TABLE 1 Preset Actions for DBMS Errors *(Continued)*

Error Number	Action	Connection State	New State	Message
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	*	*	
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	*	*	

TABLE 1 Preset Actions for DBMS Errors *(Continued)*

Error Number	Action	Connection State	New State	Message
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	STOP	*	*	Please check HA-Oracle parameters. Connect failed because target host or object does not exist
27100	STOP	*	*	Shared memory realm already exists

TABLE 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)

Index

M

N

O

P

R

S

T

U

V

W

Z