IRIS FailSafe™ Oracle®
Administrator's Guide

CONTRIBUTORS

Written by Susan Ellis
Illustrated by Dany Galgani
Edited by Christina Cary
Production by Ruth Christian
Engineering contributions by Gilberto Arnaiz, Chander Kant, and Paddy
    Sreenivasan
Cover design and illustration by Rob Aguilar, Rikk Carey, Dean Hodgkinson,
    Erik Lindholm, and Kay Maitz

Contents

List of Examples

List of Figures

List of Tables

About This Guide

This guide provides information about configuring IRIS FailSafe™ systems with the IRIS FailSafe Oracle® database option. This option enables Oracle database resources to be failed over from one node to another if a component fails. This guide is intended as a supplement to the information about configuring IRIS FailSafe included in the *IRIS FailSafe Administrator's Guide*.

This guide was prepared in conjunction with the initial release of the IRIS FailSafe Oracle option and the 1.1 Release of the IRIS FailSafe software product. It describes IRIS FailSafe Oracle software for Oracle 7.2 and Oracle 7.3.

Audience

This guide is written for system administrators who are responsible for configuring and administering an IRIS FailSafe system with the optional IRIS FailSafe Oracle software. These system administrators must be able to customize several shell scripts and must be familiar with Oracle configuration and Oracle startup and shutdown procedures.

Structure of This Document

This guide contains three chapters and an appendix:

- Chapter 1, "IRIS FailSafe Oracle Overview," describes the IRIS FailSafe components that are included in the IRIS FailSafe Oracle option. It also provides an overview on the steps required to add Oracle databases as highly available services to an IRIS FailSafe cluster.

- Chapter 2, "Configuring IRIS FailSafe for Oracle," explains the procedures for adding Oracle databases as highly available services.

- Chapter 3, "Configuration File Blocks for Oracle," contains reference information for the Oracle blocks in IRIS FailSafe configuration files.

•   Appendix A, "Messages About Configuration File Errors," explains the error messages and warnings from the *ha_cfgverify* command that are related to Oracle.

Related Documentation

For Oracle installation information, see the *Oracle7 for MIPS ABI Systems Installation and Configuration Guide*.

Besides this guide, other documentation for the IRIS FailSafe system includes

•   *IRIS FailSafe Administrator's Guide* (007-3109-xxx)

•   *IRIS FailSafe Programmer's Guide* (007-3298-xxx)

•   *IRIS FailSafe INFORMIX Administrator's Guide* (007-3268-xxx, IRIS/FailSafe INFORMIX option)

•   *IRIS FailSafe Sybase Administrator's Guide* (007-3306-xxx, IRIS FailSafe Sybase option)

The IRIS FailSafe reference pages are as follows:

•   ha_admin(1M)

•   ha_appmon(1M)

•   ha_cfgchksum(1M)

•   ha_cfginfo(1M)

•   ha_cfgverify(1M)

•   ha_exec(1M)

•   ha_hbeat(1M)

•   ha_ifa(1M)

•   ha_ifmx(1M) (IRIS FailSafe INFORMIX® option)

•   ha_killd(1M)

•   ha_nc(1M)

•   ha_orcl(1M) (IRIS FailSafe Oracle option)

•   ha_spng(1M)

•   ha_sybs(1M) (IRIS FailSafe Sybase® option)

- http_ping(1M) (IRIS FailSafe Web option)

- macconfig(1M)

- ha.conf(4)

- failsafe(7M)

Release notes are included with each IRIS FailSafe product. The names of the release notes are as follows:

ha_base   release notes for IRIS FailSafe

ha_nfs   release notes for IRIS FailSafe NFS®

ha_www   release notes for IRIS FailSafe Web

ha_ orcl   release notes for IRIS FailSafe Oracle

ha_ ifmx   release notes for IRIS FailSafe INFORMIX

ha_ sybs   release notes for IRIS FailSafe Sybase

Conventions Used In This Guide

These type conventions and symbols are used in this guide:

Bold   Literal command-line arguments and literal parameter values

Italics   Command names, filenames, new terms, the names of *inst* subsystems, manual/book titles, variable command-line arguments, and variables to be supplied by the user in examples, code, and syntax statements

`Fixed-width type`
            Examples of command output that is displayed in windows on your monitor and of the contents of files

`Bold fixed-width type`
            Commands and text that you are to type literally in response to shell and command prompts

IRIX shell prompt for the superuser (*root*)

IRIS FailSafe Oracle Overview

This chapter provides information about the components that are added to the base IRIS FailSafe product by the IRIS FailSafe Oracle option. It assumes that you are familiar with the basic components of IRIS FailSafe described in Chapter 1 of the *IRIS FailSafe Administrator's Guide*. This chapter also lists the steps you must take to add Oracle databases to the highly available services that are failed over on an IRIS FailSafe cluster.

The major sections in this chapter are as follows:

- "IRIS FailSafe Oracle Database Monitoring" on page 1
- "Database Startup and Shutdown Scripts" on page 3
- "Overview of Configuring IRIS FailSafe for Oracle" on page 3

IRIS FailSafe Oracle Database Monitoring

The IRIS FailSafe Oracle option adds a database agent, shown in Figure 1-1, to the IRIS FailSafe base software. The database agent, */usr/etc/ha_orcl*, monitors Oracle databases. If *ha_orcl* detects a failure, it takes one of two actions, based on the value of the db-avail configuration parameter:

- If db-avail is **low**, *ha_orcl* reports the failure with a message in */var/adm/SYSLOG*.
- If db-avail is **high**, *ha_orcl* exits. The application monitor detects that *ha_orcl* isn't running and reports a local monitoring failure to the node controller. Based on the states of the nodes in the cluster, the node controller determines the actions to be taken, which may include failing over the services on the node to the other node.
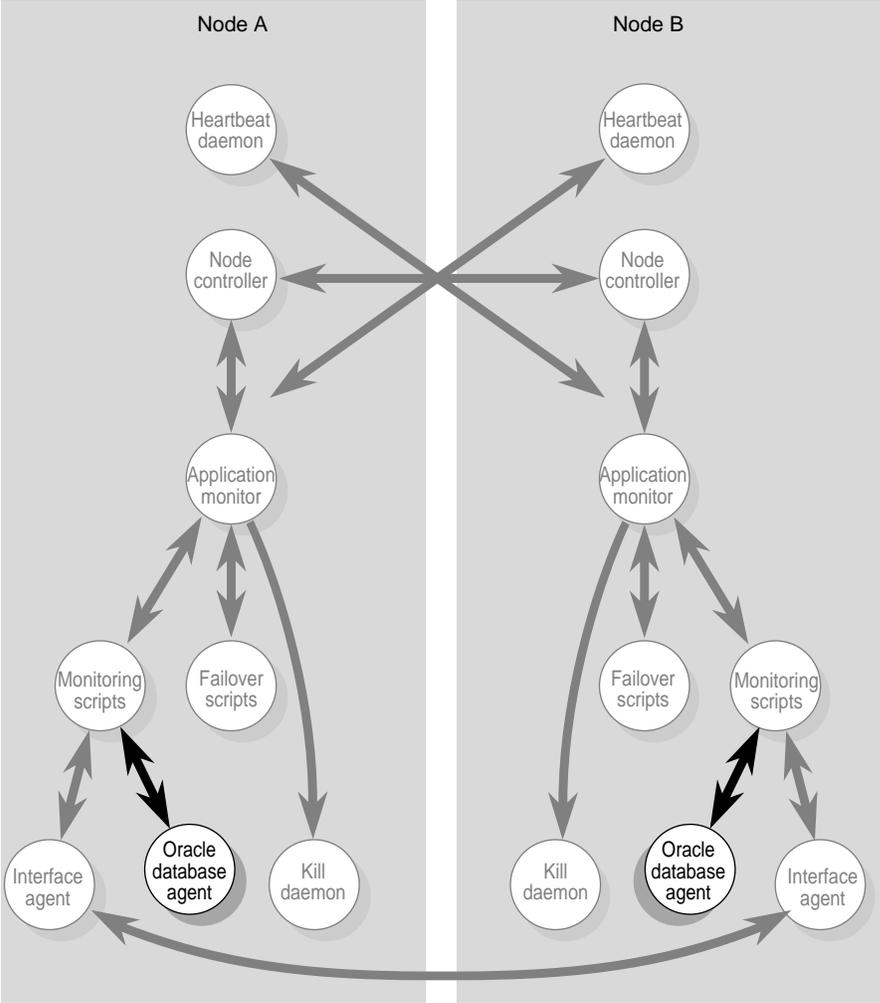
Figure 1-1   Oracle Database Agent Component in IRIS FailSafe

Database Startup and Shutdown Scripts

The standard Oracle startup and shutdown scripts, *$ORACLE_HOME/bin/dbstart* and *$ORACLE_HOME/bin/dbshut*, can't be used with IRIS FailSafe. You must use scripts that you develop by starting with the sample scripts */var/ha/samples/startup_oracle.sample* and */var/ha/samples/shutdown_oracle.sample*.

A key difference between the standard scripts and IRIS FailSafe scripts is that the standard scripts start up and shut down all Oracle databases. IRIS FailSafe scripts start up or shut down just one Oracle instance (one Oracle server accessing one Oracle database), the one identified by $ORACLE_SID.

For each type of script (startup and shutdown), you can use any combination of the following strategies:

- Write one script that takes the database instance as an argument (which IRIS FailSafe passes as an environment variable). Use it to start up (or shutdown) each database instance.

- Use different startup (or shutdown) scripts for different database instances.

- Use scripts with the same pathnames on each node, but different contents. For example, the startup script for a CHALLENGE® XL that is a primary node might be different from the startup script on a backup node that is a CHALLENGE S.

Overview of Configuring IRIS FailSafe for Oracle

To configure an IRIS FailSafe cluster for failover of Oracle databases, follow these steps:

1. Install, configure, and test the base IRIS FailSafe software as described in the *IRIS FailSafe Administrator's Guide*.

2. Install additional software if required. See the section "Required Software" in Chapter 2.

3. Choose how to configure Oracle software and databases on the disks. See the section "Planning Oracle Configuration" in Chapter 2.

4. Prepare the Oracle startup and shutdown scripts. See the section "Customizing the Oracle Startup and Shutdown Scripts" in Chapter 2.

5. Add Oracle information to the configuration file, */var/ha/ha.conf*. See the section "Adding Oracle Information to the Configuration File."

6. Install the Oracle startup and shutdown scripts and the new configuration files on each node. See the section "Installing Startup and Shutdown Scripts and the Configuration File" in Chapter 2.

7. Test Oracle failover. See the section "Testing Oracle Failover" in Chapter 2.

Configuring IRIS FailSafe for Oracle

This chapter provides information about configuring the IRIS FailSafe Oracle database option for use on an IRIS FailSafe system. The major sections in this chapter are as follows:

- "Required Software" on page 5
- "Planning Oracle Configuration" on page 6
- "Customizing the Oracle Startup and Shutdown Scripts" on page 10
- "Adding Oracle Information to the Configuration File" on page 12
- "Installing Startup and Shutdown Scripts and the Configuration File" on page 13
- "Testing Oracle Failover" on page 13

Required Software

The required software for Oracle failover is as follows:

- Oracle database software

  See the guide *Oracle7 for MIPS ABI Systems Installation and Configuration Guide*.

- Base IRIS FailSafe software

  See the section "Installing the IRIS FailSafe Software" in Chapter 2 of the *IRIS FailSafe Administrator's Guide* for a complete list of required base software.

- IRIS FailSafe Oracle software

  The software subsystem is *ha_ orcl.sw.base*.

Planning Oracle Configuration

The subsections below discuss choosing the location of key Oracle files and Oracle databases and the need for customizing certain Oracle files. They also describe an example configuration of Oracle files and the configuration parameters for this configuration.

Choosing an Oracle Configuration

In configuring Oracle on an IRIS FailSafe cluster, you need to consider the locations of these components:

- Oracle software ($ORACLE_HOME)

- The Oracle parameter file in its default location ($ORACLE_HOME/*dbs*/*init*${ORACLE_SID}.*ora* for Oracle 7.2) or a location you specify

- IRIS FailSafe versions of the Oracle startup and shutdown scripts

- Oracle databases (including all data, log, and control files)

All of the items above are specified for each Oracle instance. Therefore, you have the flexibility to use different copies of each component, for example Oracle software, for each database instance (the Oracle software used by one instance must be identical on both nodes). In addition, the Oracle parameter file, startup script, and shutdown script for a database instance need not be identical on each node. For example, if the two nodes have different amounts of memory, you may want different Oracle parameter files on each node, so that the size of the System Global Area is different on each node. The two parameter files must have identical pathnames, but the contents can be different.

To plan an IRIS FailSafe configuration for Oracle, you must answer these questions:

- What is the primary node of each database instance?

- Do all database instances use the same Oracle software or not?

- Is the Oracle parameter file identical on each node for each database instance or different?

- Are the Oracle startup and shutdown scripts identical in location and content for each database instance or different?

- For each database instance, is the content of its Oracle startup script identical on each node?

- For each database instance, is the content of its Oracle shutdown script identical on each node?

After answering all of the questions above, follow these guidelines for each database instance to decide whether to use shared or non-shared disks for each of the Oracle components for that instance:

- Oracle databases must be on one or more XFS filesystems or XLV logical volumes that are on shared disks.

- If the Oracle parameter file is identical on each node, it can be on a shared disk or replicated on non-shared disks.

- If the Oracle parameter file is different on each node, it must be on non-shared disks on each node.

- If the Oracle startup and shutdown scripts are identical on each node, they can be on a shared disk or replicated on non-shared disks.

- If the Oracle startup and shutdown scripts are different on each node, they must be on a non-shared disk on each node.

- Oracle software can be on a shared disk or replicated on non-shared disks (it must be identical on each disk).

- If the Oracle parameter file is different on each node, it must be on non-shared disks and linked to the directory in the Oracle software where Oracle expects to find it.

Once you have decided whether to store the Oracle startup and shutdown scripts on shared or non-shared disks, you can choose the pathnames of the files. Some possibilities are as follows:

- The scripts can be installed in a directory in the Oracle software, for example, */usr/oracle/scripts/dbstart*. This is a good choice when using Oracle software installed on a shared disk.

- The scripts can be installed in the directory */var/ha*, for example, */var/ha/actions/oracle_dbstart*.

7

Example Oracle Configuration

For a simple Oracle configuration with one database instance called Oracle1 whose primary node is *xfs-ha2*, an example configuration is as follows:

Oracle software
> On a shared disk with $ORACLE_HOME set to */usr/people/oracle*

Oracle parameter file
> Identical on both nodes and stored in *$ORACLE_HOME/dbs/initfs.ora*, which is on the same shared disk as the Oracle software

Oracle startup script
> Identical on both nodes and stored in *$ORACLE_HOME/scripts/dbstart*, which is on the same shared disk as the Oracle software

Oracle shutdown script
> Identical on both nodes and stored in *$ORACLE_HOME/scripts/dbshut*, which is on the same shared disk as the Oracle software

Oracle database files (data, log, and control files)
> On a filesystem that is on a shared disk and mounted at */xfs-ha2fs* on *xfs-ha2*

Configuration Parameters for Oracle

Table 2-1 lists the label and configuration parameters for the Oracle database in the configuration described in the previous section, "Example Oracle Configuration."

Table 2-1   Oracle Configuration Parameters

Label or Parameter	Example Value	Comments
database label	Oracle1	Choose any unique name.
release-dir	/usr/people/oracle	This value is the value of $ORACLE_HOME.
instance-id	fs	This value is the value of $ORACLE_SID. It is used to create the pathname of the Oracle parameter file unless a different parameter file is specified in the Oracle startup script.
startup-script	/usr/people/oracle/scripts/dbstart	The value is the location of the Oracle startup script for this instance.
shutdown-script	/usr/people/oracle/scripts/dbshut	The value is the location of the Oracle shutdown script for this instance.
master-db-fs	ofs2fs	The value is the label for one (any one) of the filesystems that is used by the database. (See the section "Oracle Blocks" in Chapter 3 for more information.) The server-node of the filesystem's volume must be the primary node of the database.

Customizing the Oracle Startup and Shutdown Scripts

Example 2-1 is a copy of the sample Oracle startup script provided in */var/ha/samples/startup_oracle.sample*. Most comments have been removed and line numbers have been added.

Example 2-1   Sample Oracle Startup Script

```
 1   #! /sbin/sh
 2
 3   PATH=$ORACLE_HOME/bin:$PATH
 4   # By default ${ORACLE_HOME}/dbs/init${ORACLE_SID}.ora will be used
 5   # used as the parameter file
 6   svrmgrl <<EOF
 7   connect internal
 8   startup
 9   EOF
10   if test $? -eq 0 ; then
11       logger "Database \"${ORACLE_SID}\" started"
12       exit 0;
13   else
14       logger "Database \"${ORACLE_SID}\" not started"
15       exit 1;
16   fi
```

As written, the script assumes that $ORACLE_HOME is set to the value of the parameter release-dir and $ORACLE_SID is set to the value of instance-id. These environment variables are set by the script */var/ha/resources/oracle*, which calls this startup script.

As mentioned in the comments on lines 4 and 5, the file *${ORACLE_HOME}/dbs/init${ORACLE_SID}.ora* is used as the Oracle parameter file by default. To use a different file, change line 7 so that it is

```
connect internal pfile=parameterfile
```

parameterfile is the full pathname of the parameter file.

Lines 6 through 9 start an Oracle instance.

Line 10 checks the exit status of the *svrmgrl* command to determine whether the database has been successfully started. In either case, a message is written to */var/adm/SYSLOG* to report the result. The script exits with a status of 0 for a successful start or 1 for an unsuccessful start.

Example 2-2 is a copy of the sample Oracle shutdown script provided in */var/ha/samples/shutdown_oracle.sample*. Again, comments have been removed and line numbers have been added.

Example 2-2   Sample Oracle Shutdown Script

```
1    #! /sbin/sh
2
3    PATH=$ORACLE_HOME/bin:$PATH
4    svrmgrl <<EOF
5    connect internal
6    shutdown normal
7    EOF
8    if test $? -eq 0 ; then
9        logger "Database \"${ORACLE_SID}\" shut down."
10       exit 0;
11   else
12       logger "Database \"${ORACLE_SID}\" not shut down."
13       exit 1;
14   fi
```

LInes 3 through 7 shut down an Oracle instance.

Line 10 checks the exit status of the *svrmgrl* command to determine whether the database has been successfully shutdown. In either case, a message is written to */var/adm/SYSLOG* to report the result. The script exits with a status of 0 for a successful shutdown or 1 for an unsuccessful shutdown.

When customizing the Oracle startup and shutdown scripts, remember these points:

• The script cannot be interactive. It must be able to run as a background process.

• The script will be called once for each oracle block in the configuration file that gives the pathname of this script as the value of startup-script or shutdown-script.

• The script is automatically run with the environment variable ORACLE_HOME set to the value of the configuration parameter release-dir and the environment variable ORACLE_SID set to the value of the configuration parameter instance-id.

• If you've changed the location of the Oracle parameters file, be sure that the startup script has the correct pathname.

Adding Oracle Information to the Configuration File

The procedure for creating a configuration file that includes Oracle configuration information is described below. The procedure assumes that a configuration file that doesn't include Oracle has been created, installed, and tested as described in the *IRIS FailSafe Administrator's Guide*. Using the procedure, you add information to a copy of the configuration file. Installing the configuration file as */var/ha/ha.conf* on both nodes is described in the section "Installing Startup and Shutdown Scripts and the Configuration File" in this chapter.

1. Copy the file */var/ha/ha.conf* to any directory on one of the nodes in the cluster.

2. Add all of the volumes that will be used for Oracle to the copy of *ha.conf* if they are not already specified in *ha.conf*. Be sure to specify the devname-owner, devname-group, and devname-mode parameters for volumes used as raw devices for Oracle database data. See the sections "Logical Volume Configuration" in Chapter 2 and "Volume Blocks" in Chapter 4 of the *IRIS FailSafe Administrator's Guide* for more information.

3. Add all of the filesystems that will be used for Oracle to the copy of *ha.conf* if they are not already specified in *ha.conf*. See the section "Filesystem Blocks" in Chapter 4 of the *IRIS FailSafe Administrator's Guide* for more information.

4. Append a copy of */var/ha/templates/ha.conf.oracle* to the end of the copy of the configuration file.

5. Complete the block called "application-class oracle." See the section "Oracle Application-Class Block" in Chapter 3 and the comments in the file for information.

6. Replicate the "oracle" block so that there is one oracle block for each database instance.

7. Using the configuration parameter choices you made in the section "Planning Oracle Configuration," and the information in the section "Oracle Blocks" in Chapter 3, prepare each of the oracle blocks.

8. Using information in the section "Oracle Blocks" in Chapter 3, prepare the "action oracle" and "action-timer oracle" blocks.

9. Check the configuration file with the *ha_cfgverify* command:

   # **/usr/etc/ha_cfgverify** *copy_filename*

   See the section "Verifying the Configuration File" in Chapter 4 of the *IRIS FailSafe Administrator's Guide* for information about the checking the configuration file with *ha_cfgverify.*

Installing Startup and Shutdown Scripts and the Configuration File

To install Oracle startup and shutdown scripts, follow these guidelines:

- If they are to be installed on a shared disk, install them from the disk's primary node.

- If they are to be installed on non-shared disks, install them on each node.

- Install the scripts in the locations specified by startup-script and shutdown-script for the Oracle database instances that use them.

- If you are using different scripts on each node, verify that the correct script is installed on each node.

- The mode of each script should be 755.

To install the new version of the configuration file */var/ha/ha.conf* on each node, you must follow the procedure described in the section "Procedure C" in Chapter 7 of the *IRIS FailSafe Administrator's Guide*.

Testing Oracle Failover

Follow this procedure to test Oracle configuration and failover:

1. Comment out all of the Oracle related blocks in */var/ha/ha.conf* on one node. Do not comment out the blocks for the filesystems and volumes used for Oracle.

2. Copy the copy of *ha.conf* from step 1 to the other node.

3. Test the failover of the volumes and filesystems that are used by Oracle and are on shared disks using the procedures in "Testing Volumes" and "Testing Filesystems" in Chapter 5 of the *IRIS FailSafe Administrator's Guide*.

4. Bring up both nodes in the cluster so that they are running IRIS FailSafe and are in normal state.

5. Bring up each Oracle database on its primary node using these commands (or the equivalent for your shell):

   ```
   # setenv ORACLE_HOME release-dir
   # setenv ORACLE_SID instance-id
   # su unix-user -c startup-script
   ```

   release-dir, *instance-id*, *unix-user*, and *startup-script* are the values of the parameters in *ha.conf* with these names for the database you are starting up. (You can identify the primary node of an Oracle database by looking at its master-db-vol or master-db-fs. The volume block for master-db-vol or the volume on which master-db-fs resides lists the server-node for the volume. This is the primary node for the database.)

6. Check that you can successfully access each of the databases using an SQL query of your choosing.

7. Shut down each Oracle database using these commands (or the equivalent for your shell) on its primary node:

   ```
   # setenv ORACLE_HOME release-dir
   # setenv ORACLE_SID instance-id
   # su unix-user -c shutdown-script
   ```

   release-dir, *instance-id*, *unix-user*, and *shutdown-script* are the values of the parameters in *ha.conf* with these names for the database you are shutting down.

8. Enter this command on each node to shut down IRIS FailSafe:

   ```
   # /etc/init.d/failsafe stop
   ```

9. Uncomment all of the Oracle blocks in */var/ha/ha.conf* on one node.

10. Copy the copy of *ha.conf* from step 9 to the other node.

11. Enter this command on each node to start up IRIS FailSafe:

    ```
    # /etc/init.d/failsafe start
    ```

12. Check that each of the databases started up and check that you can access the databases using an SQL query of your choosing on each database.

13. For the first database listed in *ha.conf* (the first oracle block), stop or kill the Oracle processes for this database. The Oracle processes are called ora_pmon_*instance-id*, ora_smon_*instance-id*, ora_dbwr_*instance-id*, and ora_lgwr_*instance-id*, where *instance-id* is the value of the instance-id parameter for the database.

14. Check that IRIS FailSafe fails over the database that you killed and all other highly available services on that node to the other node. For example, to check for Oracle processes, enter this command:

```
# ps -ef | grep ora
    root  9720 11334  0 14:58:43 pts/0    0:00 grep ora
  oracle 14936     1  0 15:26:45 ?        0:19 ora_pmon_fs
  oracle 14939     1  0 15:26:51 ?        0:03 ora_smon_fs
  oracle 14937     1  0 15:26:47 ?        0:31 ora_dbwr_fs
  oracle 14938     1  0 15:26:49 ?        0:23 ora_lgwr_fs
```

The example output shows the four processes for the database whose instance-id parameter is fs.

15. Confirm that the node on which you stopped the database is in standby state:

```
# /usr/etc/ha_admin -i
```

16. On the node that is now in standby state, enter this command to reintegrate it into the cluster:

```
# /usr/etc/ha_admin -rf
```

17. Check that the highly available services that failed over in step 14 are stopped on the backup node and restarted on their original node.

18. For each of the remaining Oracle databases listed in *ha.conf*, repeat steps 13 through 17 to verify that they fail over properly when stopped or killed.

Configuration File Blocks for Oracle

Configuration parameters for Oracle must be specified in the configuration file
/var/ha/ha.conf. The sections in this chapter describe each Oracle-specific block that must
be added and the configuration parameters within each of those blocks. The sections are
as follows:

- "Oracle Application-Class Block" on page 17
- "Oracle Blocks" on page 18
- "Oracle Action and Action-Timer Blocks" on page 21

The examples in this chapter show the Oracle configuration file blocks for the example
discussed in the sections "Example Oracle Configuration" and "Configuration
Parameters for Oracle" in Chapter 2.

Oracle Application-Class Block

Example 3-1 shows the application-class block in an Oracle configuration.

Example 3-1   Oracle Application-Class Block

```
application-class oracle
{
        server-node = xfs-ha2
        agent = /usr/etc/ha_orcl
}
```

The application-class oracle block contains these configuration parameters:

server-node   Lists a node that is a primary server for Oracle. In this example, just one
                node is listed because just one node is a primary node for Oracle. If both
                nodes are primary nodes for Oracle, server-node is listed twice.

agent   The full pathname of the database agent for Oracle. You should not
                change this parameter.

Oracle Blocks

IRIS FailSafe configuration files contain one oracle block for each Oracle database instance that is a highly available service. Example 3-2 shows an example oracle block.

Example 3-2   Oracle Block

```
oracle Oracle1
{
        master-db-fs = xfs-ha2fs
        unix-user = oracle
        release-dir = /usr/people/oracle
        instance-id = fs
        db-probe-time = 30
        db-timeout = 20
        db-avail = high
        start-monitor-time = 100
        db-retry-count = 1
        startup-script = /xfs-ha2fs/oracle/scripts/dbstart
        shutdown-script = /xfs-ha2fs/oracle/scripts/dbshut
        db-shutdown-timeout = 30
}
```
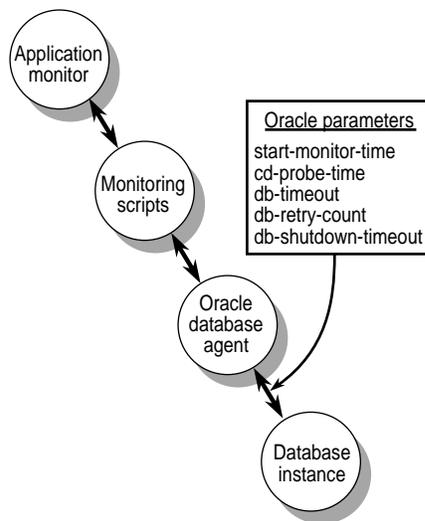
The configuration parameters used in oracle blocks are as follows:

master-db-vol

The value must match the label of a volume block. See the discussion below about master-db-vol and master-db-fs for information about which volume block label to use. This parameter or master-db-fs must be specified.

master-db-fs   The value must match the label of a filesystem block. See the discussion below about master-db-vol and master-db-fs for information about which filesystem block label to use. This parameter or master-db-vol must be specified.

unix-user   The IRIX login name that is the owner of the Oracle release directory.

release-dir   The Oracle release directory specified in the Oracle configuration. This value is also the value of the environment variable ORACLE_HOME. This parameter is required.

instance-id   The value of $ORACLE_SID for the Oracle instance that IRIS FailSafe is monitoring. This parameter is required.

18

db-probe-time

Defines the length of time (in seconds) between the completion of one probe of the database by the Oracle database agent and the beginning of the next probe. This parameter is required.

db-timeout   Defines the time the Oracle database agent waits for a response to its probe from the Oracle database instance in seconds. This parameter is required.

db-avail   The possible values are **high** and **low**. If the value is **high**, a database server failure forces a failover. If the value is **low**, a failure of the database server doesn't force a failover, but the failover is reported. The default value is **high**.

start-monitor-time

The amount of time that the Oracle database agent waits after receiving a message to start monitoring before it performs its first probe of the Oracle database instance. This wait ensures that the database instance has had time to start up.

db-retry-count The number of monitoring retries by the database agent before a failure is declared. If db-retry-count isn't specified, its value defaults to **1**.

startup-script

The pathname of a custom startup script that replaces the standard Oracle startup script. See the section "Customizing the Oracle Startup and Shutdown Scripts" in Chapter 2 for more information. This parameter is required.

shutdown-script

The pathname of a custom shutdown script that replaces the standard Oracle shutdown script. See the section "Customizing the Oracle Startup and Shutdown Scripts" in Chapter 2 for more information. This parameter is required.

db-shutdown-timeout

The timeout for the Oracle shutdown script specified by shutdown-script. If the shutdown script doesn't complete in this many seconds, IRIS FailSafe performs an abort shutdown of the Oracle instance.

The Oracle system tablespace files could reside in multiple filesystems and on multiple logical volumes. You must choose one of these filesystems or volumes to be the "master" and specify it as the value of master-db-fs or master-db-vol. The oracle blocks do not list all filesystems and volumes that are part of the database, just the filesystem or volume containing system tablespace files and designated as the master. However, the configuration file must list all volumes and filesystems used by each database in volume and filesystem blocks (see the sections "Volume Blocks" and "Filesystem Blocks" in Chapter 2 of the *IRIS FailSafe Administrator's Guide* for information on these blocks). The IRIS FailSafe system uses the volume and filesystem blocks as a specification of the shared storage that must be online prior to bringing up the database.

Figure 3-2 shows the monitoring frequency parameters in each oracle block. They control the monitoring of a database instance by the Oracle database agent.

Figure 3-1   Monitoring Frequency Configuration Parameters in Oracle Blocks

Table 3-1 lists each of these monitoring frequency parameters and their suggested, minimum, and maximum values. It also describes relationships between different parameters and the rounding that occurs for some parameter values.

Table 3-1   Values for Monitoring Frequency Configuration Parameter in Oracle Blocks

Parameter	Suggested Value	Minimum Value	Maximum Value	Comments
start-monitor-time (oracle block)	100			Rounded to the nearest ten seconds.
db-probe-time	30			Rounded to the nearest ten seconds.
db-timeout	20			Rounded to the nearest ten seconds.
db-retry-count	1			
db-shutdown-timeout	30		One half of the long-timeout value	Rounded to the nearest ten seconds.

Oracle Action and Action-Timer Blocks

Example 3-3 shows the action and action-timer blocks for Oracle. The action block specifies the pathnames of the local monitoring script and the action-timer block specifies monitoring timing and timeout values for the monitoring of the Oracle database agent by the application monitor.

Example 3-3   Oracle Action and Action-Timer Blocks

```
action oracle
{
        local-monitor = /var/ha/actions/ha_orcl_lmon
}

action-timer oracle
{
        start-monitor-time = 300
        lmon-probe-time = 60
        lmon-timeout = 30
        retry-count = 2
}
```

The parameters used in action and action-timer blocks for Oracle are as follows:

local-monitor   The pathname of the local monitoring script for Oracle. Do not change this value.

start-monitor-time

Specifies the amount of time that the application monitor waits before it starts using the local monitoring script to monitor the Oracle database agent. The value of this parameter should be greater than or equal to the largest value specified for start-monitor parameters in any of the oracle blocks and it should be greater than or equal to the value of long-timeout.

lmon-probe-time

Local monitoring of the Oracle agent is done this often (in seconds).

lmon-timeout

Local monitoring of the Oracle agent times out in this many seconds if no response is received.

retry-count   Specifies the number of times the local monitoring script retries its probes of the Oracle database agent. This values doesn't affect the lmon-timeout value.

Figure 3-2 shows the monitoring parameters in the "action-timer oracle" block. They control the monitoring of the Oracle database agent by the local monitoring script.
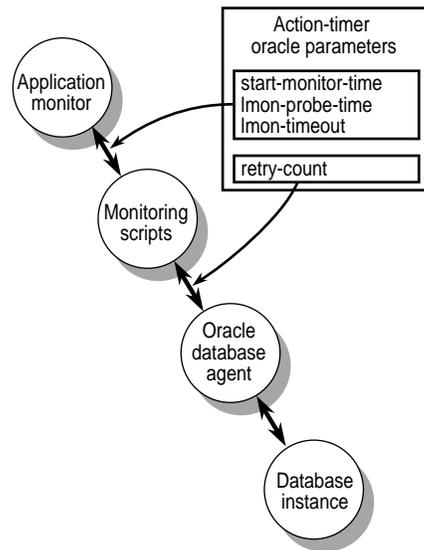
Figure 3-2   Monitoring Configuration Parameters in the Action-Timer Oracle Block

Table 3-2 lists each of these monitoring parameters and their suggested, minimum, and maximum values. It also describes relationships between different parameters and the rounding that occurs for some parameter values.

Table 3-2   Values for Monitoring Frequency Configuration Parameter in Oracle Action-Timer Blocks

Parameter	Suggested Value	Minimum Value	Maximum Value	Comments
start-monitor-time (action-timer block)	300			Should be greater than or equal to the value of long-timeout and greater than or equal to the largest of the values specified for start-monitor-time in any of the oracle blocks.
lmon-probe-time	60			
lmon-timeout	30			
retry-count	2			This value doesn't affect the lmon-timeout value.

Messages About Configuration File Errors

This appendix lists the error messages and warnings that can appear as output of the *ha_cfgverify* command that are related to Oracle configuration. Other messages are described in Appendix A of the *IRIS FailSafe Administrator's Guide*.

Note: *ha_cfgverify* generates warnings when a possible error exists, but it is unable to determine for sure. You must check each warning manually. A warning could indicate invalid IRIS FailSafe configuration.

The error messages are given in alphabetical order.

```
ha_cfgverify: No oracle section present in the file
```

> There is an Oracle application-class entry but there is no oracle block.

```
ha_cfgverify: oracle <Oracle block label> : the master-db-fs entry
<filesystem> is invalid
```

> The master-db-fs entry must match the label of a filesystem block.

```
ha_cfgverify: oracle <Oracle block label> : the master-db-vol entry <xlv
volume> is invalid
```

> The master-db-vol entry must match the label of a volume block.

```
ha_cfgverify: oracle <Oracle block label> section must have db-probe-time
entry
ha_cfgverify: oracle <Oracle block label> section must have
db-retry-count entry
ha_cfgverify: oracle <Oracle block label> section must have
db-shutdown-timeout entry
ha_cfgverify: oracle <Oracle block label> section must have db-timeout
entry
ha_cfgverify: oracle <Oracle block label> section must have either
master-db-vol or master-db-fs entry
ha_cfgverify: oracle <Oracle block label> section must have instance-id
entry
ha_cfgverify: oracle <Oracle block label> section must have release-dir
entry
ha_cfgverify: oracle <Oracle block label> section must have
shutdown-script entry
ha_cfgverify: oracle <Oracle block label> section must have
start-monitor-time entry
ha_cfgverify: oracle <Oracle block label> section must have
startup-script entry
ha_cfgverify: oracle <Oracle block label> section must have unix-user
entry
```

A mandatory parameter is missing from the oracle block.

```
ha_cfgverify: oracle <Oracle block label> section: unix-user entry <user
name> is invalid
```

The system failed to get a password file entry for the user specified as unix-user.

```
ha_cfgverify: Warning: oracle <Oracle block label> section: release-dir
entry <directory> is invalid
```

The specified directory is not accessible or is not a directory or the underlying filesystem may not be mounted.

```
ha_cfgverify: Warning: oracle <Oracle block label> section:
shutdown-script <script path> is invalid
ha_cfgverify: Warning: oracle <Oracle block label> section:
startup-script <script path> is invalid
```

The specified file is not accessible or the underlying filesystem may not be mounted.