



**SIOS Protection Suite for Linux  
Oracle Recovery Kit  
v9.0.2**

**Administration Guide**

**Mar 2016**

This document and the information herein is the property of SIOS Technology Corp. (previously known as SteelEye® Technology, Inc.) and all unauthorized use and reproduction is prohibited. SIOS Technology Corp. makes no warranties with respect to the contents of this document and reserves the right to revise this publication and make changes to the products described herein without prior notification. It is the policy of SIOS Technology Corp. to improve products as new technology, components and software become available. SIOS Technology Corp., therefore, reserves the right to change specifications without prior notice.

LifeKeeper, SteelEye and SteelEye DataKeeper are registered trademarks of SIOS Technology Corp.

Other brand and product names used herein are for identification purposes only and may be trademarks of their respective companies.

To maintain the quality of our publications, we welcome your comments on the accuracy, clarity, organization, and value of this document.

Address correspondence to:  
[ip@us.sios.com](mailto:ip@us.sios.com)

Copyright © 2015  
By SIOS Technology Corp.  
San Mateo, CA U.S.A.  
All rights reserved

# Table of Contents

---

|   |           |
|---|-----------|
| <b>Chapter 1: Introduction</b> .....                                | <b>1</b>  |
| Document Contents .....   | 1         |
| SIOS Protection Suite Documentation .....                           | 1         |
| <b>Requirements</b> .....   | <b>2</b>  |
| Kit Hardware and Software Requirements .....                        | 2         |
| <b>Chapter 2: Configuring Oracle with LifeKeeper</b> .....          | <b>3</b>  |
| Specific Configuration Considerations for Oracle .....              | 3         |
| Configuring the Oracle Net Listener for LifeKeeper Protection ..... | 7         |
| Listener Configuration .....  | 7         |
| Configuring Transparent Application Failover with LifeKeeper .....  | 9         |
| Configuration Examples .....  | 11        |
| Configuration Requirements .....                                    | 11        |
| Active/Standby Configurations .....                                 | 12        |
| Active/Active Configurations .....                                  | 15        |
| <b>Chapter 3: LifeKeeper Configuration Tasks</b> .....              | <b>20</b> |
| Creating an Oracle Resource Hierarchy .....                         | 21        |
| Deleting a Resource Hierarchy .....                                 | 22        |
| Extending Your Hierarchy .....                                      | 25        |
| Unextending Your Hierarchy .....                                    | 28        |
| <b>Viewing Oracle Configuration Settings</b> .....                  | <b>30</b> |
| Changing Username / Password for the Oracle Database Account .....  | 31        |
| Testing Your Resource Hierarchy .....                               | 33        |
| Performing a Manual Switchover from the GUI .....                   | 33        |
| Recovery Operations .....   | 33        |
| <b>Chapter 4: Troubleshooting</b> .....                             | <b>35</b> |

---

|  |           |
|--|-----------|
| <b>Oracle Known Issues and Restrictions</b> .....  | <b>35</b> |
| Control File Switchover Failure .....  | 35        |
| Truncated Output .....   | 35        |
| Flash Recovery Destination Located on a Shared Drive .....                                 | 35        |
| Prevent Failover When Unable to Connect to the Database .....                              | 36        |
| Non-Traditional Location for oratab .....  | 36        |
| NFS Version 4 Not Supported .....  | 37        |
| Creating Oracle Database Hierarchy After Installing Oracle Binaries on Local Storage ..... | 37        |
| Oracle Listener Stays in Service on Primary Server After Failover .....                    | 39        |
| Database Creation Problems .....   | 39        |
| Database Startup Problems .....  | 39        |
| inqfail error in the LifeKeeper Log .....  | 40        |
| <b>Appendix</b> .....  | <b>41</b> |
| Setting Up Oracle to Use Raw I/O .....   | 41        |
| Adding a Tablespace After Creating Hierarchy .....   | 43        |
| Creating an Oracle Listener for Multiple Resources .....                                   | 43        |
| Updating the Protected Listener(s) .....   | 47        |

# Chapter 1: Introduction

## Oracle Recovery Kit Documentation

The SIOS Protection Suite for Linux Oracle® Recovery Kit provides fault resilience for Oracle software in an SPS environment. The Recovery Kit software furnishes a mechanism to tie the data integrity of Oracle databases to the increased availability provided by SIOS Protection Suite.

### Document Contents

This documentation contains the following topics:

SPS for Linux Technical Documentation (also available from the Help menu within the LifeKeeper GUI). A list of all the SPS for Linux documentation and where the information is available.

[Requirements](#). Before you can install and set up the recovery software, your server must meet certain hardware and software requirements. You should refer to the SIOS Protection Suite Installation Guide for specific instructions on how to install or remove the LifeKeeper Oracle Recovery Kit.

[Configuring Your Recovery Kit](#). To ensure that your SPS configuration provides the protection and flexibility you require, you need to be aware of the configuration rules. To appropriately plan your configuration, you must understand your network configuration, interface selection, user system setup, hierarchy options and the Oracle configuration tasks. In addition to planning your configuration, this section also includes configuration examples and the specific tasks required to configure your Recovery Kit.

[Troubleshooting](#). This section provides a list of informational and error messages with recommended solutions.

### SIOS Protection Suite Documentation

The following SPS product documentation is available from the SIOS Technology Corp. website:

- SPS for Linux Release Notes
- SPS for Linux Technical Documentation
- Optional Recovery Kit Documentation

This documentation, along with documentation associated with optional LifeKeeper Recovery Kits, is provided on the SIOS Technology Corp. website at:

<http://docs.us.sios.com/>

## Requirements

Before attempting to install or remove the LifeKeeper Oracle Recovery Kit, you must understand the hardware and software requirements and the installation and removal procedures.

### Kit Hardware and Software Requirements

Be sure that your configuration meets the following requirements:

**Servers.** The Recovery Kit requires two or more supported computers configured in accordance with LifeKeeper requirements described in the SPS Technical Documentation and the SPS Release Notes.

**Shared Storage.** Oracle databases must reside on a shared disk in an SPS environment. Depending on your shared storage architecture, the appropriate LifeKeeper shared storage or multipath storage kit will need to be installed on each node in your cluster. In the example of NFS backed database storage, installation of the LifeKeeper NAS Kit is necessary.

**LifeKeeper Software.** You must install the same version of LifeKeeper software and any patches on each server. Please refer to the SPS Release Notes and SPS Technical Documentation for specific LifeKeeper requirements.

**LifeKeeper IP Recovery Kit.** This Recovery Kit is required if remote clients will be accessing the Oracle Database. You must have the same version of this Recovery Kit on each server.

**IP Network Interface.** Each server requires at least one Ethernet TCP/IP-supported network interface. In order for IP switchover to work properly, user systems connected to the local network should conform to standard TCP/IP specifications.

**Note:** Even though each server requires only a single network interface, you should use multiple interfaces for a number of reasons: heterogeneous media requirements, throughput requirements, elimination of single points of failure, network segmentation, and so forth.

**TCP/IP Software.** Each server also requires the TCP/IP software.

**Oracle Software.** Each server must have the Oracle software installed and configured before you can configure LifeKeeper and the LifeKeeper Oracle Recovery Kit. The same version should be installed on each server. Consult the SPS Release Notes or your sales representative for the latest release compatibility and ordering information.

You should refer to the SPS Installation Guide for specific instructions on how to install or remove the LifeKeeper Oracle Recovery Kit.

## Chapter 2: Configuring Oracle with LifeKeeper

This section contains information you should consider before you start to configure Oracle and examples of typical Oracle configurations.

The following tasks are described in the Administration section within the SPS for Linux Technical Documentation because they are common tasks with steps that are identical across all Recovery Kits.

- **Create a Resource Dependency.** Creates a parent/child dependency between an existing resource hierarchy and another resource instance and propagates the dependency changes to all applicable servers in the cluster.
- **Delete a Resource Dependency.** Deletes a resource dependency and propagates the dependency changes to all applicable servers in the cluster.
- **In Service.** Brings a resource hierarchy into service on a specific server.
- **Out of Service.** Takes a resource hierarchy out of service on a specific server.
- **View/Edit Properties.** View or edit the properties of a resource hierarchy on a specific server.

For instructions on installing Oracle on Linux distributions using the 2.6 kernel, please see your Linux distribution's website.

Also, please refer to your SPS for Linux Technical Documentation located on the SIOS Technology website for instructions on configuring your LifeKeeper Core resource hierarchies.

### Specific Configuration Considerations for Oracle

**Note:** If you plan to use Oracle with Raw I/O, the Raw I/O devices must be properly set up prior to use. See the [Appendix](#) for instructions. (Raw I/O is not an option for LifeKeeper Single Server Protection.)

Before configuring the Oracle Recovery Kit, complete the following preparatory steps to ensure that file systems and disk partitions used by Oracle will be accessible from each server.

1. **Remove personal initialization file prompts.** For the Oracle Recovery Kit to work properly, you must remove (or comment out) all prompts in the personal initialization file (i.e., `.profile`, `.bash_profile`) for the Oracle user. This file is specific to the shell that is being used by the Oracle user. The file cannot be interactive.  
**Note:** If “`stty`” statements are going to be in the personal initialization file, they must be in an “if” statement that verifies that an interactive terminal is being used.
2. **Configure Kernel Parameters.** Please refer to the Oracle documentation for information on how linux kernel parameters such as shared memory and other kernel resources should be configured. An example of how to set these parameters is below.

On *each server* in the cluster:

- a. Set the following `ipcs` limits in `/etc/sysctl.conf` before configuring LifeKeeper.

```
# changes for Oracle
kernel.shmmax = <value>
kernel.shmmni = <value>
kernel.shmall = <value>
kernel.sem = <value>
```

- b. Run `sysctl -p` to set the above changes in the kernel.
- c. On certain distributions you may need to add `sysctl -p` to the system initialization file (i.e. `boot.local` or `rc.local`) so that these kernel changes are set after each reboot.

3. **\$ORACLE\_HOME directory.** When you configure the `$ORACLE_HOME` directory and associated files on local disks, be sure that the `$ORACLE_HOME` directory and files are identical on all servers. Use the standard Linux utilities to create and copy directories and files to the set of servers.  
**Note:** In certain active/active configurations, the location of `$ORACLE_HOME` are different.
4. **Location.** The `$ORACLE_HOME` directory can be on shared or non-shared disks. The advantage to having the directory on shared media is that you only need to configure files such as the parameter file `Oracle_HOME/dbs/<initSID.ora or spfileSID.ora>` once, if the same shared disk is used for `$ORACLE_HOME` (e.g. in an active/standby configuration). The disadvantage to the shared directory is that direct access to the file system is available to only one server at a time. SCSI reservations permit only one server at a time access to a LifeKeeper protected shared drive. If creating an active-active cluster configuration where two or more Oracle instances (SID) will be protected independently in the cluster, `$ORACLE_HOME` must be installed on local, non-shared storage.
5. **User and Group ID.** An oracle user (oracle) and group (dba) should be created on all servers. The user ID and group ID numbers must be the same on all servers.
6. **Databases, archive files, log files and control files.** All databases, archive files, log files, and control files must be created on shared file systems or disk partitions. These locations are set in the Oracle parameter file `init<SID>.ora` or `spfileSID.ora`. Please refer to the Oracle documentation for information on editing database parameters. The pathnames must be the same for all servers. Oracle internally keeps this information in its control file; therefore, SYSTEMS database space and paths cannot be changed unless Oracle is running.

**Note:** Oracle log archiving is not enabled by default. If it is enabled prior to the creation of the LifeKeeper Oracle hierarchy, LifeKeeper will detect the location of the archive files and create a separate file system hierarchy if necessary. But if log archiving is enabled after the LifeKeeper Oracle hierarchy has been created, you must manually create and extend a file system hierarchy to protect the shared archive location, and create a dependency from the Oracle resource to this new file system hierarchy.

**Note:** When using storage applications with locking and following recommendations for the NFS mount options, SPS requires the additional `nolock` option be set, e.g.  
`rw,nolock,bg,hard,nointr,tcp,nfsvers=3,timeo=600,rsize=32768,wsiz=32768,actimeo=0.`

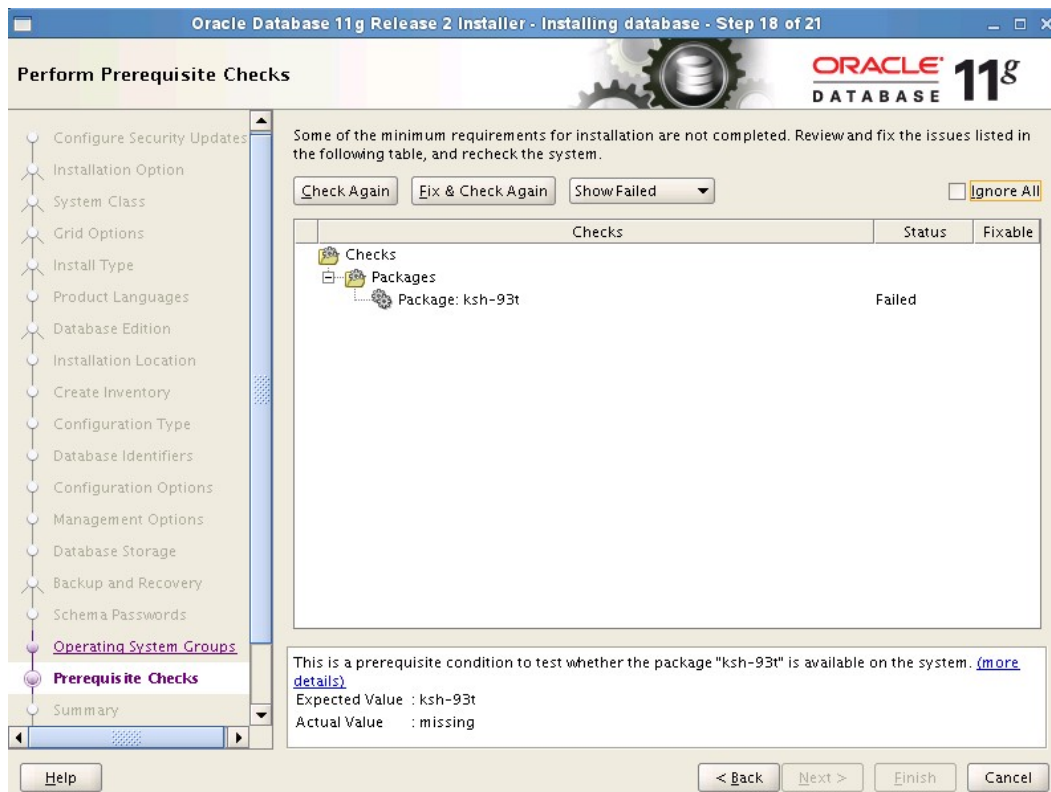


- On a new installation of Oracle, the final configuration of the database instance is easier if the database installation program is not allowed to create a database. When the installer asks if you want to create a database, select **No**. After the installation is complete, run the **Oracle Database Creation Assistant (dbca)**. dbca provides much better control of where database components get created. When running dbca, specify that the Flash Recovery Area gets created on LifeKeeper protected storage (this applies to Oracle 11g).

**Important:** The Flash Recovery Destination must be located on a shared drive.

If runInstaller is allowed to create a database, the Flash Recovery Area will have to be relocated manually. (**Note:** Allowing runInstaller to create a database is not recommended.)

- During the installation of Oracle using the “runInstaller” utility, there will be a point where the installer verifies the packages and configuration of Linux before proceeding with the Oracle database installation. If LifeKeeper 7.2 (or higher version) has already been installed, a message complaining about a missing ksh package will appear.



If this message is displayed, check the box in the upper righthand corner, **Ignore All**. The installation of LifeKeeper has removed the ksh package and replaced it with the Public Domain Korn Shell, `pdksh`. Oracle should install fine using `pdksh`.

**Note:** Beginning with Version 8.0, the Oracle ARK no longer requires `pdksh`; however, `pdksh` is still required by the LifeKeeper core and therefore still requires checking the **Ignore All** setting.

**Note:** Beginning with Version 8.1, LifeKeeper provides its own private `pdksh` package and therefore does not conflict with Oracle's `ksh` requirements.

9. Tune the database engine. Refer to Oracle documentation for guidelines on tuning the database engine for data integrity and performance. In particular, the tuning for memory caching and checkpointing frequency is critical to optimizing the application for fault resilience. The checkpointing interval determines the number of database transactions that have not been committed to disk and therefore would be lost during a system failure.
10. Database entry in `oratab` file. The `/etc/oratab` file must contain an entry for the database. The LifeKeeper configuration routines use the contents of this file to relate `$ORACLE_HOME` and `$ORACLE_SID` values. Usually, the Oracle installation program creates the required entry. In a configuration in which the Oracle software is installed to a shared file system, however, you must copy the `oratab` file from the server where the Oracle installation was performed to the `/etc` directory of the other servers so that it is available to all the servers.

**Note:** The configuration can have only one `oratab` per server. Refer to the Oracle Product Manual for information on the file format.

**Note:** The `oratab` file can be accommodated in other locations besides `/etc`. By default, the Oracle ARK looks for the `oratab` file in `/etc` followed by `/var/opt/oracle`. If the `oratab` file is not located in one of these default locations, then `ORACLE_ORATABLOC` must be set in `/etc/default/LifeKeeper` to the directory containing `oratab`.

11. Disable automatic start-ups. Since LifeKeeper is responsible for starting the databases it controls, be sure to disable any automatic start-up actions. LifeKeeper disables automatic start-up when a hierarchy is created. This is accomplished by modifying the `oratab` file.
12. The Listener configuration file, `listener.ora`. New lines should not be embedded in the entries (e.g., `SID_NAME=xx` should be on one line).
13. Oracle Database Username and Password. LifeKeeper will use local session and OS Authentication to control Oracle Database. If you would like to turn off local OS Authentication for security reasons, LifeKeeper can use the specified username and password. The Oracle Database user must be able to connect as `sysdba` authority to the database to be protected, and each server's Oracle Database must have the same username and password. If this configuration is skipped during resource creation, then LifeKeeper will not use username and password to control the Oracle Database resource. This parameter can be added, changed or removed any time after creating the resource.

Once under LifeKeeper protection, the LifeKeeper and database user privileges can be lowered from `sysdba` to `sysoper`. See [Changing Username / Password for the Oracle Database Account](#) for more information.

### Tips for Creating the Oracle Username and Password.

- a. On the node where the Oracle database is running, log in to Linux with a user that is part of the `dba` group. (The "oracle" account is most common.) Using the `sqlplus` utility, connect to the database as the administrative user by issuing the following command:

```
$ sqlplus / as sysdba
```

- b. Create a new user for this function:

```
SQL> CREATE USER lkdba IDENTIFIED BY
      "password";
```

- c. Then grant this user SYSDBA privileges:

```
SQL> GRANT SYSDBA to lkdba;
```

- d. If Oracle has been configured so that each node in the LifeKeeper cluster has a local copy of \$ORACLE\_HOME, execute these commands on each node in the cluster. After creating the LifeKeeper Oracle hierarchy, bring the database in service on the node and then execute the `CREATE` and `GRANT` commands (above) to set up the user in Oracle.

**CAUTION:** Avoid configuring two databases on the same file system. If you must configure two databases on the same file system, exercise great care. In this situation, both databases must be placed under LifeKeeper protection and both hierarchies must have the same primary and backup servers.

## Configuring the Oracle Net Listener for LifeKeeper Protection

If your Oracle database will have remote client connections, you will want to protect the Oracle Listener in addition to the Oracle database server. Please refer to the Oracle documentation for information on using Oracle network configuration utilities to create Oracle network configuration files such as `listener.ora` and `tnsnames.ora`.

**Note:** Refer to the [Creating a Shared Oracle Listener for Multiple Resources](#) section in the appendix in this document for instructions on how to create a shared Oracle Listener for multiple resources.

### Listener Configuration

1. You need to choose a switchable IP address for clients to make connections to. You may want to put this address in DNS. (Refer to the **LifeKeeper IP Recovery Kit Documentation** for details on creating an IP resource hierarchy. Refer to the topic **Creating a Resource Dependency** under GUI Administration Tasks for details on creating a resource dependency.)
2. In the `listener.ora` file, specify this switchable IP address as the HOST for the database service name. (See the Oracle documentation for details about the `listener.ora` file.) Although the DNS name can be used in place of the switchable IP address for the HOST database service name, LifeKeeper best practices does not recommend this. Using the switchable IP address will prevent DNS lookup issues from impacting LifeKeeper's ability to determine the status of a running listener during quickCheck, restore or remove processing. Additionally, a `SID_LIST_LISTENER` stanza must be defined, even though you may have only one listener defined.

Sample format of a `listener.ora`:

```
.
.
.
SID_LIST_LISTENER =
```

## Listener Configuration

```
(SID_LIST =
(SID_DESC =
      (SID_NAME = <SID Name>)
)
)
.
.
.

<listener name>=

(DESCRIPTION_LIST =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP) (HOST = <switchable IP>) (PORT =
      <port number>))
    )
  )
)
.
.
.
```

3. Specify the switchable IP address as the HOST in the `tnsnames.ora` file or Oracle Names. (See the Oracle documentation for details about the `tnsnames.ora` file.) Although the DNS name can be used in place of the switchable IP address for the HOST database service name, LifeKeeper best practices does not recommend this. Using the switchable IP address will prevent DNS lookup issues from impacting LifeKeeper's ability to determine the status of a running listener during quickCheck, restore or remove processing:

```
.
.
.
<SID Name>=
(DESCRIPTION =
  (ADDRESS_LIST =
    (ADDRESS = (PROTOCOL = TCP) (HOST = <switchable IP>) (PORT =
      <port number>))
    )
  (CONNECT_DATA =
    (SID = <SID Name>)
  )
)
```

These sample files should work with both Oracle 10g and 11g:

```
listener.ora

SID_LIST_LISTENER =
(SID_LIST =
(SID_DESC =
```

## Configuring Transparent Application Failover with LifeKeeper

```
(SID_NAME = ORA11A)
)
)

LISTENER =
  (DESCRIPTION_LIST =
    (DESCRIPTION =
      (ADDRESS = (PROTOCOL = TCP) (HOST = 192.0.2.0) (PORT = 1521))
    )
  )

tnsnames.ora

ORA01 =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP) (HOST = 192.0.2.0) (PORT = 1521))
    )
    (CONNECT_DATA =
      (SID = ORA01)
    )
  )
```

The normal location of `listener.ora` is in `$ORACLE_HOME/network/admin`. The most common port number is 1521. The global name of the database was defined at creation time. Also keep in mind, if the `$ORACLE_HOME` directory is installed on non-shared storage, a copy of `listener.ora` will need to be on both systems.

**Note:** Oracle Net provides the option of automatically failing over client connections to another listener if the listener for a service should fail. To take advantage of this feature, set the `FAILOVER` parameter to “**ON**” in the `tnsnames.ora` file. If the listener for the LifeKeeper-protected Oracle SID should fail, this allows client connections to continue through another listener until LifeKeeper recovers the protected listener.

## Configuring Transparent Application Failover with LifeKeeper

When a server failover or an Oracle database failure occurs, users can be severely disrupted. Typically the user’s connections to the database will be lost along with most work in progress. Upon the completion of the failover (or recovery of the Oracle database), clients will have to restart their application and reconnect to the database. With the Transparent Application Failover (TAF) feature of Oracle, this disruption can be reduced or eliminated by masking some types of failures. To configure TAF in a LifeKeeper environment, there are tasks that must be performed on both the LifeKeeper server side and the Oracle client side.

For clients to effectively take advantage of the TAF feature, the client application must use failover-aware API calls from the Oracle Call Interface (OCI). The clients must also configure the appropriate TAF support using the Oracle Net parameters in the `tnsnames.ora` file. TAF mode can be configured by including a `FAILOVER_MODE` parameter under the `CONNECT_DATA` section of the `tnsnames.ora` connect descriptor. The TAF mechanism supports several sub-parameters to control and affect the behavior of a client connection during failover. The LifeKeeper for Linux Oracle Recovery Kit supports the following TAF configuration sub-parameters:

**TYPE= (SELECT or SESSION).**

This value determines how TAF will handle client connection failover. When the type is set to **SELECT**, Oracle keeps track of all select statements issued during transition. Upon establishment of a new connection, the select statements are re-executed, and the cursors repositioned so clients can continue to fetch rows. When type is set to **SESSION** only a new connection is created; work in progress may be lost.

**METHOD= (BASIC).**

With this method TAF will attempt a reconnect only after the primary connection fails. The alternative method is **PRECONNECT**, LifeKeeper does not currently support the use of **PRECONNECT** as a method.

**DELAY= (#sec).**

This value is the number of seconds that TAF will wait between attempts to connect following a failure. This value should be carefully determined for your client application and environment.

**RETRIES= (#number of tries).**

This value is the number of times that TAF will attempt to retry a failed connection before giving up. The combination of **DELAY** and **RETRIES** must allow enough time for a complete recovery of Oracle in the event of a server failure. This will give TAF enough time to restart after the server failover has completed.

An excerpt from a sample tnsnames.ora file for a client system is included below.

```
LKproDB=
  (DESCRIPTION=
    (ADDRESS_LIST=
      (ADDRESS = (PROTOCOL=TCP) (HOST=<switchableIP>)
        (PORT=<port number>))
    )
    (CONNECT_DATA=
      (SID=LKroDB)
      (SERVER=DEDICATED)
      (FAILOVER_MODE=
        (TYPE=SELECT)
        (METHOD=BASIC)
        (DELAY=5)
        (RETRIES=30)
      )
    )
  )
)
```

The normal location of `tnsnames.ora` is in `$ORACLE_HOME/network/admin`. The most common port number is 1521. “`tnsnames.ora`” files can also be located in user’s home directories as well. Also keep in mind, if the `$ORACLE_HOME` directory has been installed on non-shared storage, a copy of `listener.ora` and `tnsnames.ora` will need to be on both systems.

On the LifeKeeper server protecting the Oracle database, the listener should be configured using a LifeKeeper-protected switchable IP address. Refer to the [Configuring the Oracle Net Listener for LifeKeeper Protection](#) section above for details on configuring Oracle Net and listener support.

## Configuration Examples

The following figures illustrate examples of both active/standby and active/active Oracle configurations in an SPS environment.

The examples in this section show how Oracle database instances can be configured on local and shared disks. Each diagram shows the relationship between the type of configuration and the Oracle parameters. Each configuration also adheres to the configuration rules and requirements described in this administration guide that ensure compatibility between the Oracle configuration and the LifeKeeper software.

This section first describes the configuration requirements and then provides these configuration examples:

- [Active/Standby](#)
- [Active/Active](#)

The examples in this section are only a sample of the configurations you could establish, but understanding these configurations and adhering to the configuration rules helps you define and set up workable solutions for your computing environment.

## Configuration Requirements

Each of the examples involves one or two databases: **databaseA** and **databaseB**. By default, LifeKeeper offers a tag name matching the Oracle database system identifier (SID). However, the screen examples in the following pages use tag names consisting of the SID and server name such as `databaseA-on-server1`.

To understand the configuration examples, keep these configuration requirements in mind:

- **LifeKeeper hierarchy.** When performing LifeKeeper administration, the primary server refers to the location where the Oracle instance is currently running. System administration takes place on this server when creating a LifeKeeper hierarchy. For the configuration examples, the primary server is Server 1 and the backup or alternate server is Server 2.
- **Shared disk locked by one server.** When shared storage resources are under LifeKeeper protection, they can only be accessed by one server at a time. If the shared device is a disk array, an entire LUN is protected. If a shared device is a disk, then the entire disk is protected. This prevents inadvertent corruption of the data by other servers in the cluster. When a server fails, the highest priority backup server establishes its own protection, locking out all other servers.
- **Database on shared disk.** In order for the LifeKeeper Oracle Recovery Kit to function properly, the

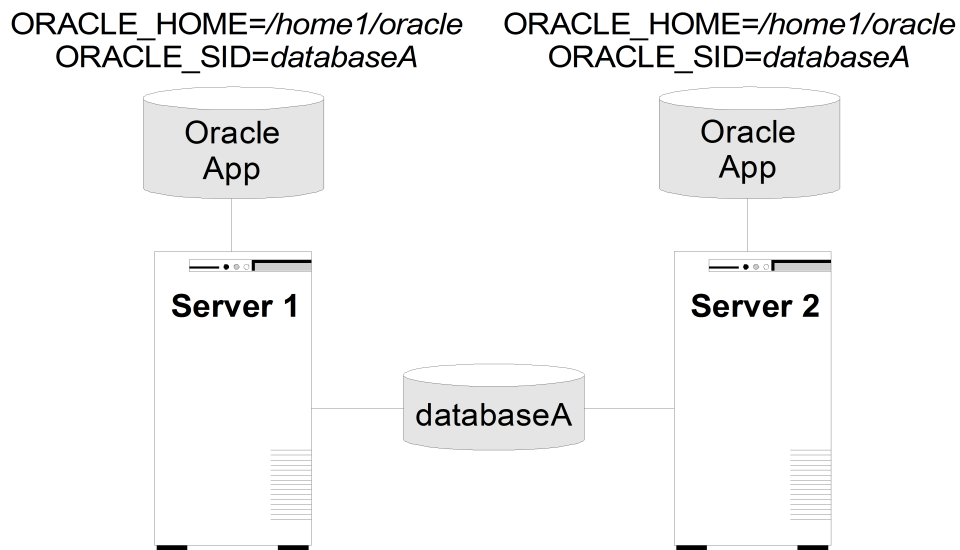
database must always be on a shared device. The database may be on one or more file systems and/or disks.

**Note:** The tables associated with each configuration illustration provide examples of the appropriate information that would be entered in the Create Resource Hierarchy wizard on the primary server (Server 1) and Extend Resource Hierarchy wizard to the backup server (Server 2). For additional detail on what information to enter into the wizards, refer to the [“LifeKeeper Configuration Tasks”](#) section. These tables can be a helpful reference when configuring your Recovery Kit.

## Active/Standby Configurations

This section provides two active/standby configuration examples, shown in Figure 1 and Figure 2. In these configurations, Server 1 is considered active because it has exclusive access to the database. Server 2 does other processing. If Server 1 fails, Server 2 gains access to the database and LifeKeeper re-establishes the database operations.

**Figure 1. Active/Standby Configuration, Example 1**



### Configuration Notes:

1. Each server has its own \$ORACLE\_HOME directory on a non-shared disk. Each server has the same version of the Oracle application.
2. The \$ORACLE\_HOME path is the same on both servers.
3. The database, databaseA, is on a shared disk.

### Creating a resource hierarchy on Server 1:

|                          |           |
|--------------------------|-----------|
| Server:                  | Server1   |
| ORACLE_SID for Database: | databaseA |



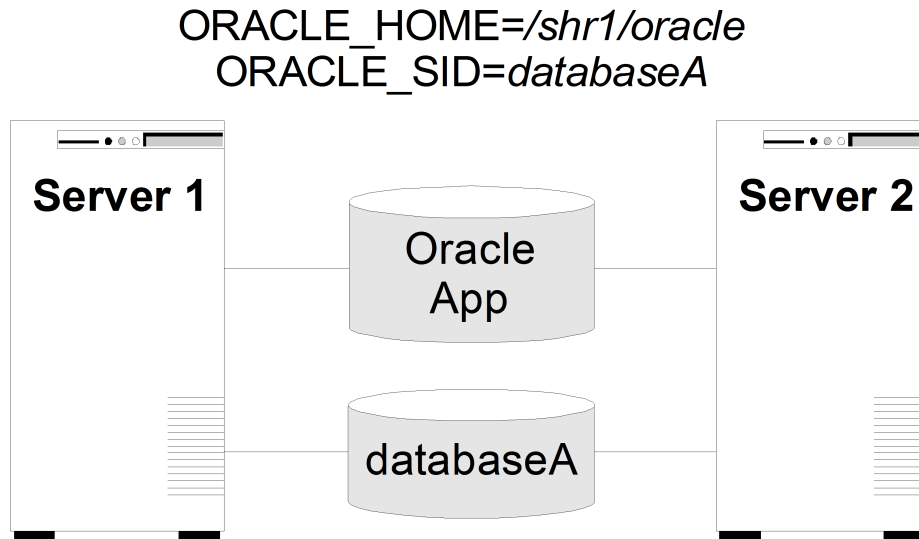
## Active/Standby Configurations

|                           |                      |
|---------------------------|----------------------|
| Username for Database     | system               |
| Password for Username     | *****                |
| ORACLE_HOME for Database: | /home1/oracle        |
| Database Tag:             | databaseA-on-server1 |

### Extending the resource hierarchy to Server 2:

|                  |                      |
|------------------|----------------------|
| Template Server: | Server1              |
| Tag to Extend    | databaseA-on-server1 |
| Target Server    | Server2              |
| Target Priority: | 10                   |
| Database Tag:    | databaseA-on-server1 |

Figure 2. Active/Standby Configuration, Example 2

**Configuration Notes:**

1. Both servers use the \$ORACLE\_HOME directory on a shared disk.
2. The \$ORACLE\_HOME path is the same on both servers.
3. The database, databaseA, is on a shared disk.
4. Server 2 can not access files and directories on the shared disk while Server 1 is active.
5. \$ORACLE HOME can be on the same shared disk as the database or on separate disks.

**Creating a resource hierarchy on Server 1:**

|                           |                      |
|---------------------------|----------------------|
| Server:                   | Server1              |
| ORACLE_SID for Database:  | databaseA            |
| Username for Database     | system               |
| Password for Username     | *****                |
| ORACLE_HOME for Database: | /shr1/oracle         |
| Database Tag:             | databaseA-on-server1 |

**Extending the resource hierarchy to Server 2:**

|                  |                      |
|------------------|----------------------|
| Template Server: | Server1              |
| Tag to Extend    | databaseA-on-server1 |
| Target Server    | Server2              |
| Target Priority: | 10                   |
| Database Tag:    | databaseA-on-server1 |

## Active/Active Configurations

An active/active configuration consists of at least two servers, each running a different database instance. The databases must be on different shared disks.

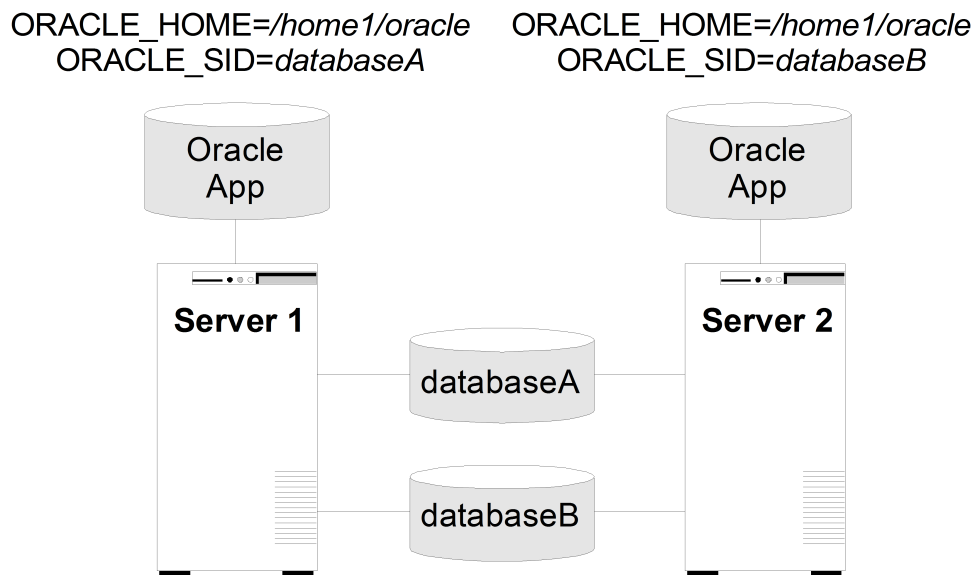
\$ORACLE\_HOME can be on non-shared or on shared disks depending upon the configuration requirements. For example, multiple database instances on any of the servers using a common \$ORACLE\_HOME require \$ORACLE\_HOME to be on non-shared disks. If the \$ORACLE\_HOME directories are on shared disk, they must be on separate shared disks.

This section provides two active/active configuration examples, shown in Figure 3 and Figure 4:

- Databases on shared resources and a common \$ORACLE\_HOME on non-shared resources.
- Databases on shared resources and the appropriate \$ORACLE\_HOME instance on the same shared resource.

**Note:** Multiple database instances on one server using multiple instances of \$ORACLE\_HOME on non-shared resources are not illustrated.

**Figure 3. Active/Active Configuration, Example 1**



**Configuration Notes:**

1. The server has its own \$ORACLE\_HOME directory on a non-shared disk. Each server has the same version of the Oracle application.
2. The \$ORACLE\_HOME path is the same on both servers.
3. The databases, databaseA and databaseB, are on shared disks.
4. The oratab file exists in /etc/ on both servers, containing entries for both Oracle instances.
5. Initially, Server 1 runs databaseA and Server 2 runs databaseB. In a switchover situation, one system can run both databases.
6. See [Creating Oracle Database Hierarchy After Installing Oracle Binaries on Local Storage](#) for further information.

**Creating the first resource hierarchy on Server 1:**

|                           |                      |
|---------------------------|----------------------|
| Server:                   | Server1              |
| ORACLE_SID for Database:  | databaseA            |
| Username for Database     | system               |
| Password for Username     | *****                |
| ORACLE_HOME for Database: | /home1/oracle        |
| Database Tag:             | databaseA-on-server1 |

**Extending the first resource hierarchy to Server 2:**

|                  |                      |
|------------------|----------------------|
| Template Server: | Server1              |
| Tag to Extend    | databaseA-on-server1 |
| Target Server    | Server2              |
| Target Priority: | 10                   |
| Database Tag:    | databaseA-on-server1 |

**Creating a second resource hierarchy on Server 2:**

|                          |           |
|--------------------------|-----------|
| Server:                  | Server2   |
| ORACLE_SID for Database: | databaseB |
| Username for Database    | system    |

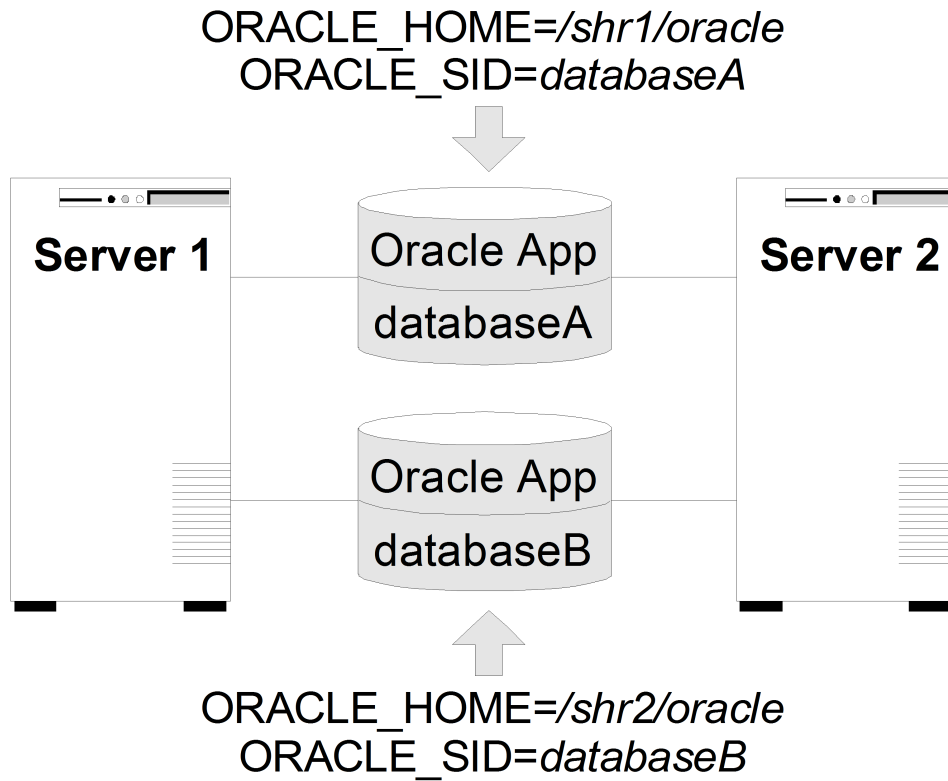
Active/Active Configurations

|                           |                      |
|---------------------------|----------------------|
| Password for Username     | *****                |
| ORACLE_HOME for Database: | /home1/oracle        |
| Database Tag:             | databaseB-on-server2 |

**Extending the second resource hierarchy to Server 1:**

|                  |                      |
|------------------|----------------------|
| Template Server: | Server2              |
| Tag to Extend    | databaseB-on-server2 |
| Target Server    | Server1              |
| Target Priority: | 10                   |
| Database Tag:    | databaseB-on-server2 |

Figure 4. Active/Active Configuration, Example 2



**Configuration Notes:**

## Active/Active Configurations

1. Both servers use an \$ORACLE\_HOME directory on different shared disks.
2. The Oracle application is the same on both servers. The \$ORACLE\_HOME is different for each instance defined on the server.
3. The databases, databaseA and databaseB, are on shared disks.
4. The oratab file exists in /etc/, containing entries for both Oracle instances.
5. A unique login is required for each Oracle instance. The id and gid for each login should be the same on Server 1 and Server 2.
6. Initially, Server 1 runs databaseA and Server 2 runs databaseB. In a switchover situation, one system can run both databases.

### Creating the first resource hierarchy on Server 1:

|                           |                      |
|---------------------------|----------------------|
| Server:                   | Server1              |
| ORACLE_SID for Database:  | databaseA            |
| Username for Database     | system               |
| Password for Username     | *****                |
| ORACLE_HOME for Database: | /shr1/oracle         |
| Database Tag:             | databaseA-on-server1 |

### Extending the first resource hierarchy to Server 2:

|                  |                      |
|------------------|----------------------|
| Template Server: | Server1              |
| Tag to Extend    | databaseA-on-server1 |
| Target Server    | Server2              |
| Target Priority: | 10                   |
| Database Tag:    | databaseA-on-server1 |

### Creating a second resource hierarchy on Server 2:

|                          |           |
|--------------------------|-----------|
| Server:                  | Server2   |
| ORACLE_SID for Database: | databaseB |
| Username for Database    | system    |
| Password for Username    | *****     |

## Active/Active Configurations

|                           |                      |
|---------------------------|----------------------|
| ORACLE_HOME for Database: | /shr2/oracle         |
| Database Tag:             | databaseB-on-server2 |

### Extending the second resource hierarchy to Server 1:

|                  |                      |
|------------------|----------------------|
| Template Server: | Server2              |
| Tag to Extend    | databaseB-on-server2 |
| Target Server    | Server1              |
| Target Priority: | 10                   |
| Database Tag:    | databaseB-on-server2 |

## Chapter 3: LifeKeeper Configuration Tasks

You can perform the following configuration tasks from the LifeKeeper GUI. The following four tasks are described in this guide, as they are unique to an Oracle resource instance and different for each Recovery Kit.

- [Create a Resource Hierarchy](#). Creates an application resource hierarchy in your LifeKeeper cluster.
- [Delete a Resource Hierarchy](#). Deletes a resource hierarchy from all servers in your LifeKeeper cluster.
- [Extend a Resource Hierarchy](#). Extends a resource hierarchy from the primary server to a backup server.
- [Unextend a Resource Hierarchy](#). Unextends (removes) a resource hierarchy from a single server in the LifeKeeper cluster.
- [View Oracle Configuration Settings](#) - Allows viewing of the Resource Properties dialog.
- [Change Username / Password](#). Change the Username and Password to login to protect Oracle Database.

The following tasks are described in the Administration section within the SPS for Linux Technical Documentation because they are common tasks with steps that are identical across all Recovery Kits.

- [Create a Resource Dependency](#). Creates a parent/child dependency between an existing resource hierarchy and another resource instance and propagates the dependency changes to all applicable servers in the cluster.
- [Delete a Resource Dependency](#). Deletes a resource dependency and propagates the dependency changes to all applicable servers in the cluster.
- [In Service](#). Brings a resource hierarchy into service on a specific server.
- [Out of Service](#). Takes a resource hierarchy out of service on a specific server.
- [View/Edit Properties](#). View or edit the properties of a resource hierarchy on a specific server.

**Note:** Throughout the rest of this section, we explain how to configure your Recovery Kit by selecting certain tasks from the **Edit** menu of the LifeKeeper GUI. You can also select each configuration task from the toolbar. You may also right-click on a global resource in the Resource Hierarchy Tree (left-hand pane) of the status display window to display the same drop down menu choices as the Edit menu. This, of course, is only an option when a hierarchy already exists.

You can also right-click on a resource instance in the Resource Hierarchy Table (right-hand pane) of the status display window to perform all the configuration tasks, except *Creating a Resource Hierarchy*, depending on the state of the server and the particular resource.



## Creating an Oracle Resource Hierarchy

**Note:** In order to take advantage of Oracle Net remote client access, the IP address used for client connectivity must be under LifeKeeper protection as a dependent of the Oracle hierarchy. (Refer to the section [Configuring the Oracle Net Listener for LifeKeeper Protection](#) for details.)

To create a resource instance from the primary server, you should complete the following steps:

1. From the LifeKeeper GUI menu, **select** Edit, then **Server**. From the drop down menu, select **Create Resource Hierarchy**.

**Important:** The Oracle Application must be running when you create the resource.

A dialog box will appear with a drop down list box with all recognized Recovery Kits installed within the cluster. Select **Oracle Database** from the drop down listing. Click **Next** to proceed to the next dialog box.

**Note:** When the **Back** button is active in any of the dialog boxes, you can go back to the previous dialog box. This is especially helpful should you encounter an error that might require you to correct previously entered information.

If you click **Cancel** at any time during the sequence of creating your hierarchy, LifeKeeper will cancel the entire creation process.

2. Select the **Switchback Type**. This dictates how the Oracle instance will be switched back to this server when it comes back into service after a failover to the backup server. You can choose either intelligent or automatic. Intelligent switchback requires administrative intervention to switch the instance back to the primary/original server. Automatic switchback means the switchback will occur as soon as the primary server comes back on line and reestablishes LifeKeeper communication paths.

The switchback type can be changed later, if desired, from the **General** tab of the **Resource Properties** dialog box. Click **Next** to proceed to the next dialog box.

3. Select the **Server** where you want to place the Oracle Database (typically this is referred to as the primary or template server). All the servers in your cluster are included in the drop down box. Click **Next** to proceed to the next dialog box.
4. Select the **ORACLE\_SID** for the Database ID. This is the tag name that specifies the Oracle system identifier of the database being configured. An entry for this database must exist in */etc/oratab*. Click **Next** to proceed to the next dialog box.
5. Input the **Username for ORACLE\_SID**. This is the Oracle Database Username specified during login to ORACLE\_SID. This username must be able to connect as sysdba authority to the database to gain full control. Click **Next** to proceed to the next dialog box. (This field can be left empty. If left empty, LifeKeeper will not use Username and Password to control the Oracle Database resource, and the next step, **Input Password**, will be skipped.)
6. Input **Password**. This is the password specified during login to ORACLE\_SID. The password will be saved by LifeKeeper with encrypting. Click **Next** to proceed to the next dialog box.
7. Select the **tag name** of the Listener to be included as a dependency of the Oracle resource. The list

displays all the currently protected Listener resource(s) on the server. Select the Listener resource tag that corresponds to the required listener(s) for the Oracle SID. Select **None** if no Listener resource exists.

8. Select or enter the **Database Tag**. This is a tag name that LifeKeeper gives to the Oracle hierarchy. You can select the default or enter your own tag name.

When you click **Create**, the **Create Resource Wizard** will create your Oracle resource.

9. At this point, an information box appears and LifeKeeper will validate that you have provided valid data to create your Oracle resource hierarchy. If LifeKeeper detects a problem, an ERROR will appear in the information box. If the validation is successful, your resource will be created.

```
Creating database/oracle resource...
Tue Feb 28 13:38:42 EST 2012 burn create[17114]: INFO:RKBase:create::000000:BEGIN
create of "ORA" on server "burn"
Tue Feb 28 13:38:49 EST 2012 burn create[17114]: INFO:oracle:create::122516:Creating
dependency between Oracle database "ORA (JEF)" and the dependent resource "/oracle"
on "burn".
Tue Feb 28 13:38:49 EST 2012 burn create[17114]:
INFO:oracle:create::122522:Performing in-service of new Oracle resource tag=< ORA >
on "burn".
Tue Feb 28 13:38:49 EST 2012 burn create[17114]: INFO:RKBase:create::000000:END
successful create of "ORA" on server "burn"
```

Click **Next** to proceed to the **Pre-extend dialog box** which is explained later in this documentation. You must extend the hierarchy to another server in your cluster in order to place it under LifeKeeper protection.

## Deleting a Resource Hierarchy

To delete a resource hierarchy from all the servers in your SPS environment, complete the following steps:

1. From the LifeKeeper GUI menu, select **Edit**, then **Resource**. From the drop down menu, select **Delete Resource Hierarchy**.
2. Select the name of the **Target Server** where you will be deleting your Oracle resource hierarchy.

**Note:** If you selected the Delete Resource task by right-clicking from the right pane on an individual resource instance, or from the left pane on a global resource where the resource is on only one server this dialog box will not appear.

## Deleting a Resource Hierarchy

A screenshot of a dialog box with a light beige background. On the left, the text "Target Server" is followed by a text input field containing the word "burn". To the right of the input field is a small square button with a downward-pointing triangle, indicating a dropdown menu.

Target Server

Click **Next** to proceed to the next dialog box.

3. Select the Hierarchy to Delete. Identify the resource hierarchy you wish to delete, and highlight it.

**Note:** If you selected the Delete Resource task by right clicking from either the left pane on a global resource or the right pane on an individual resource instance, this dialog will not appear.

A screenshot of a dialog box with a light beige background. On the left, the text "Hierarchy to Delete" is displayed. To the right is a large white rectangular area representing a list of resource hierarchies. The top of this area has a blue header bar with the text "ORA" in white. The rest of the list area is currently empty.

Hierarchy to Delete

ORA

Click **Next** to proceed to the next dialog box.

4. An information box appears confirming your selection of the target server and the hierarchy you have selected to delete.

You have specified the following resource hierarchy for deletion.  
Target Server: burn  
Target Tags:  
ORA

Click **Delete** to delete your resource and proceed to the final dialog box.

5. Another information box appears confirming that the Oracle resource was deleted successfully.

```
Deleting resource hierarchy ORA
Removing root resource hierarchy starting at "ORA":
Tue Feb 28 14:33:16 EST 2012 burn
/opt/LifeKeeper/subsys/database/resources/oracle/actions/delete[24962]:
INFO:RKBase:/opt/LifeKeeper/subsys/database/resources/oracle/actions/delete::000000:
BEGIN delete of "ORA" on server "burn"
Tue Feb 28 14:33:24 EST 2012 burn
/opt/LifeKeeper/subsys/database/resources/oracle/actions/delete[24962]:
INFO:RKBase:/opt/LifeKeeper/subsys/database/resources/oracle/actions/delete::000000:
END successful delete of "ORA" on server "burn"
Tue Feb 28 14:33:20 EST 2012 wake1
/opt/LifeKeeper/subsys/database/resources/oracle/actions/delete[26543]:
INFO:RKBase:/opt/LifeKeeper/subsys/database/resources/oracle/actions/delete::000000:
BEGIN delete of "ORA" on server "wake1"
Tue Feb 28 14:33:20 EST 2012 wake1
/opt/LifeKeeper/subsys/database/resources/oracle/actions/delete[26543]:
INFO:RKBase:/opt/LifeKeeper/subsys/database/resources/oracle/actions/delete::000000:
END successful delete of "ORA" on server "wake1"
Tue Feb 28 14:33:28 EST 2012 wake1 delete[26653]: INFO:RKBase:delete::000000:BEGIN
delete of "datarep-oracle" on server "wake1"
Tue Feb 28 14:33:33 EST 2012 wake1 delete[26653]: INFO:RKBase:delete::000000:END
successful delete of "datarep-oracle" on server "wake1"
Tue Feb 28 14:33:36 EST 2012 burn delete[25280]: INFO:RKBase:delete::000000:BEGIN
delete of "datarep-oracle" on server "burn"
Tue Feb 28 14:33:37 EST 2012 burn delete[25280]: INFO:RKBase:delete::000000:END
successful delete of "datarep-oracle" on server "burn"
Successfully removed
```

6. Click **Done** to exit out of the **Delete Resource Hierarchy** menu selection.

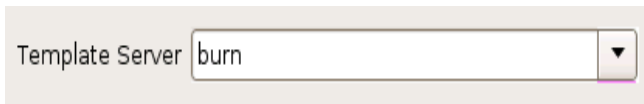
**Note:** Refer to the [Creating a Shared Oracle Listener for Multiple Resources](#) section in the appendix of this document for instructions on how to create a shared Oracle Listener for multiple resources.

## Extending Your Hierarchy

After you have created a hierarchy, you will want to extend that hierarchy to another server in the cluster. There are three possible scenarios to extend your resource instance from the template server to a target server. The first scenario is when you “**Continue**” from creating the resource into extending that resource to another server by clicking **Next** on the information dialog box displayed at the completion of the create. The second scenario is when you enter the **Extend Resource Hierarchy** task from the edit menu as shown below. The third scenario is when you right-click on an unextended hierarchy in either the left- or right-hand pane. Each scenario takes you through the same dialog boxes (with a few exceptions, which are clearly detailed below).

1. If you are entering the **Extend Wizard** from the LifeKeeper GUI menu, select **Edit**, then **Resource**. From the drop-down menu, select **Extend Resource Hierarchy**. This will launch the **Pre-Extend Resource Hierarchy Wizard**.
2. The first dialog box to appear will ask you to select the **Template Server** where your Oracle resource hierarchy is currently in service. It is important to remember that the **Template Server** you select now and the **Tag to Extend** that you select in the next dialog box represent an in service resource hierarchy. An error message will appear if you select a resource tag that is not in service on the template server you selected. The drop-down box in this dialog provides the names of all the servers in your cluster.

**Note:** If you are entering the **Pre-Extend Resource Hierarchy** task immediately following the creation of an Oracle resource hierarchy, this dialog box will not appear, since the wizard has already identified the template server in the create stage. This is also the case when you right-click on either the Oracle resource icon in the left hand pane or right-click on the Oracle resource box in the right hand pane the of the GUI window and choose **Extend Resource Hierarchy**.



It should be noted that if you click the **Cancel** button at any time during the sequence of extending your hierarchy, LifeKeeper will cancel the extension process to that particular server. However, if you have already extended the resource to another server, that instance will continue to be in effect until you specifically unextend it.

For example, let’s say you have created your resource on Server 1 and extended that resource to Server 2. In the middle of extending the same resource to Server 3, you change your mind and click on the **Cancel** button inside one of the dialog boxes. This will cancel only your action to extend the resource to Server 3, not the extension you created to Server 2. If you want to remove Server 2 from this hierarchy, you must unextend the resource from Server 2.

Click **Next** to proceed to the next dialog box.

3. Select the **Tag to Extend**. This is the name of the Oracle instance you wish to extend from the template server to the target server. The wizard will list in the drop-down box all the resources that you have created on the template server, which you selected in the previous dialog box.

**Note:** Once again, if you are entering the Pre-Extend Resource Hierarchy task immediately following the creation of an Oracle resource hierarchy, **this dialog box will not appear**, since the wizard has already identified the tag name of your Oracle resource in the create stage. This is also the case when you right-click on either the Oracle resource icon in the left hand pane or on the Oracle resource box in the right hand pane of the GUI window and choose **Extend Resource Hierarchy**.



A screenshot of a GUI dialog box. It features a label 'Tag to Extend' followed by a text input field containing the text 'ORA'. To the right of the input field is a small downward-pointing triangle icon, indicating a dropdown menu.

Click **Next** to proceed to the next dialog box.

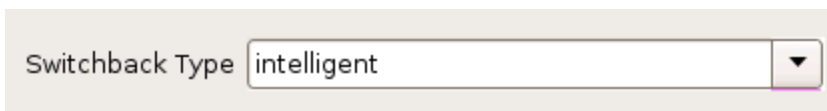
4. Select the **Target Server** where you are extending your Oracle resource hierarchy. The drop-down box provides the names of the servers in your cluster that are not already in the selected hierarchy.



A screenshot of a GUI dialog box. It features a label 'Target Server' followed by a text input field containing the text 'wake1'. To the right of the input field is a small downward-pointing triangle icon, indicating a dropdown menu.

Click **Next** to proceed to the next dialog box.

5. Select the **Switchback Type**. This dictates how the Oracle instance will be switched back to this server when it comes back into service after a failover to the backup server. You can choose either *intelligent* or *automatic*. Intelligent switchback requires administrative intervention to switch the instance back to the primary/original server. Automatic switchback means the switchback will occur as soon as the primary server comes back on line and reestablishes LifeKeeper communication paths.



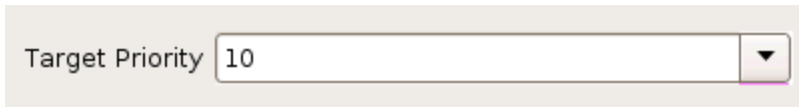
A screenshot of a GUI dialog box. It features a label 'Switchback Type' followed by a text input field containing the text 'intelligent'. To the right of the input field is a small downward-pointing triangle icon, indicating a dropdown menu.

The switchback type can be changed later, if desired, from the General tab of the Resource Properties dialog box.

Click **Next** to proceed to the next dialog box.

6. Select or enter a **Template Priority**. This is the priority for the Oracle hierarchy on the server where it is currently in service. Any unused priority value from 1 to 999 is valid, where a lower number means a higher priority (1=highest). The extend process will reject any priority for this hierarchy that is already in use by another system. The default value is recommended. Note: This selection will appear only for the initial extend of the hierarchy.

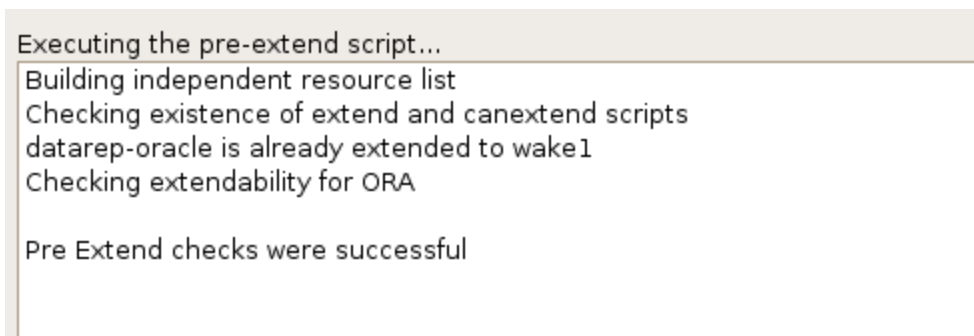
7. Select or enter the **Target Priority**. This is the priority for the new extended Oracle hierarchy relative to equivalent hierarchies on other servers. Any unused priority value from 1 to 999 is valid, indicating a server's priority in the cascading failover sequence for the resource. A lower number means a higher priority (1=highest). Note that LifeKeeper assigns the number "1" to the server on which the hierarchy is created by default. The priorities need not be consecutive, but no two servers can have the same priority for a given resource.



A screenshot of a web form element. It consists of a label 'Target Priority' followed by a text input field containing the number '10'. To the right of the input field is a small downward-pointing arrow icon, indicating a dropdown menu.

Click **Next**.

8. An information box will appear explaining that LifeKeeper has successfully checked your environment and that all the requirements for extending this Oracle resource have been met. If there were some requirements that had not been met, LifeKeeper would not allow you to select the Next button, and the Back button would be enabled.



```
Executing the pre-extend script...
Building independent resource list
Checking existence of extend and canextend scripts
datarep-oracle is already extended to wake1
Checking extendability for ORA

Pre Extend checks were successful
```

If you click **Back**, you can make changes to your resource extension according to any error messages that may appear in the information box.

If you click **Cancel** now, you will need to come back and extend your Oracle resource hierarchy to another server at some other time to put it under LifeKeeper protection.

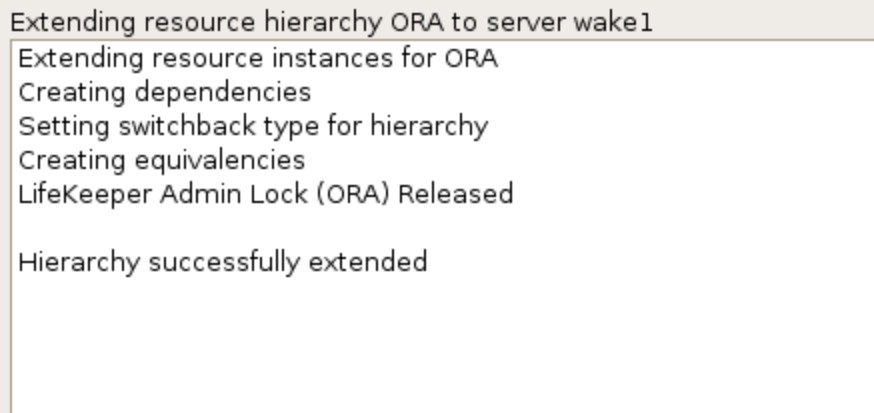
When you click **Next**, LifeKeeper will launch you into the Extend Resource Hierarchy configuration task.

9. Once the **Database Tag** displays, click **Extend**.



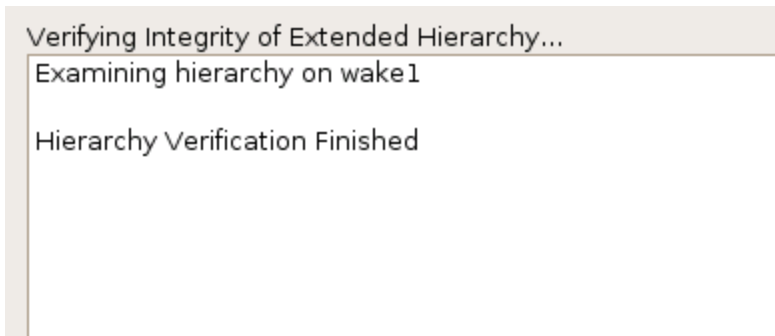
A screenshot of a web form element. It consists of a label 'Database Tag' followed by a text input field containing the text 'ORA'.

10. An information box will appear verifying that the extension is being performed.



Click **Next Server** if you want to extend the same Oracle resource instance to another server in your cluster. This will repeat the **Extend Resource Hierarchy** operation.

If you click **Finish**, another dialog box will appear confirming LifeKeeper has successfully extended your Oracle resource.



11. Click **Done** to exit from the **Extend Resources Hierarchy** menu selection.

**Note:** Be sure to test the functionality of the new instance on both servers.

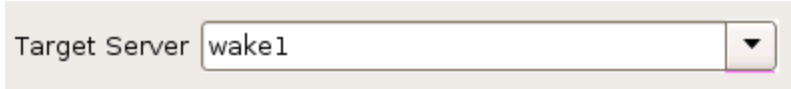
## Unextending Your Hierarchy

1. From the **LifeKeeper GUI** menu, select **Edit**, then **Resource**. From the drop-down menu, select **Unextend Resource Hierarchy**.
2. Select the **Target Server** where you want to unextend the Oracle resource. It cannot be the server where Oracle is currently in service.

**Note:** If you selected the **Unextend** task by right-clicking from the right pane on an individual resource instance, this dialog box will not appear.




## Unextending Your Hierarchy



Target Server

Click **Next** to proceed to the next dialog box.

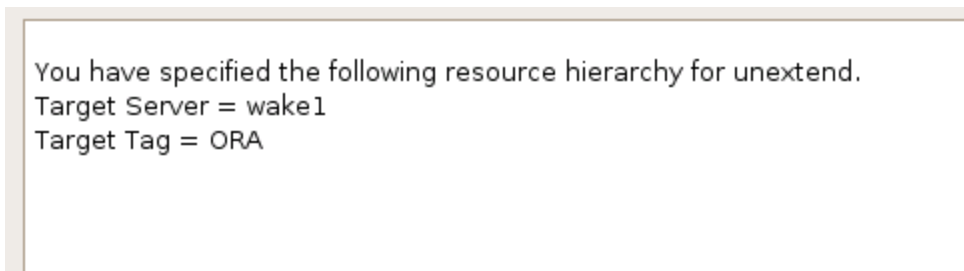
3. Select the **Oracle Hierarchy to Unextend**. **Note:** If you selected the **Unextend** task by right-clicking from either the left pane on a global resource or the right pane on an individual resource instance, this dialog will not appear.



Hierarchy to Unextend

Click **Next** to proceed to the next dialog box.

4. An information box appears confirming the target server and the Oracle resource hierarchy you have chosen to unextend.



You have specified the following resource hierarchy for unextend.  
Target Server = wake1  
Target Tag = ORA

Click **Unextend**.

5. Another information box appears confirming that the Oracle resource was unextended successfully.

```

Unextending resource hierarchy ORA from wake1
Hierarchy Unextend Manager Initializing
Checking Target Machine Communication Paths
LifeKeeper Admin Lock Flag (ORA) Established
Removing Equivalencies
Removing Resources and Associated Dependencies
Tue Feb 28 14:07:29 EST 2012 wake1 delete[10086]: INFO:RKBase:delete::000000:BEGIN
delete of "datarep-oracle" on server "wake1"
mdadm: stopped /dev/md0
Tue Feb 28 14:07:33 EST 2012 wake1 delete[10086]:
INFO:RKBase:delete::000000:stopping the monitor for /dev/md0 ...
Tue Feb 28 14:07:33 EST 2012 wake1 delete[10086]:
INFO:RKBase:delete::000000:stopping the nbd-client for /dev/nbd0 ...
Tue Feb 28 14:07:33 EST 2012 wake1 delete[10086]:
INFO:RKBase:delete::000000:/dev/md0 has been stopped
Tue Feb 28 14:07:33 EST 2012 wake1 delete[10086]: INFO:DR:delete::104015:The mirror
"/dev/md0" (resource: "datarep-oracle") has been successfully removed from system
"wake1"
Tue Feb 28 14:07:34 EST 2012 wake1 delete[10086]: INFO:DR:delete::104019:The mirror
for resource "datarep-oracle" has been successfully unextended from system "wake1"
Tue Feb 28 14:07:34 EST 2012 wake1 delete[10086]: INFO:RKBase:delete::000000:END
successful delete of "datarep-oracle" on server "wake1"
LifeKeeper Admin Lock Flag (ORA) Released
Synchronizing LifeKeeper Databases
Unextend completed successfully

```

6. Click **Done** to exit out of the **Unextend Resource Hierarchy** menu selection.

## Viewing Oracle Configuration Settings

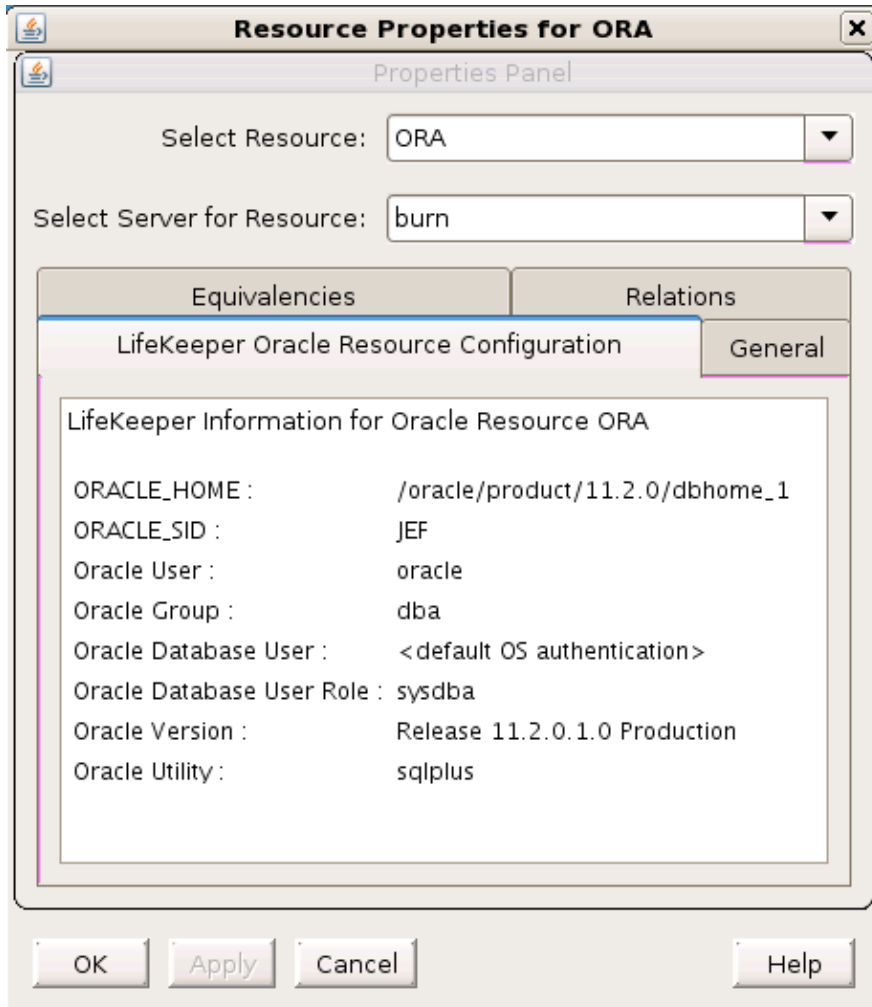
The **Resource Properties** dialog is available from the **Edit** menu or from a resource context menu. This dialog displays the properties for a particular resource on a server. When accessed from the **Edit** menu, you can select the resource and the server. When accessed from a resource context menu, you can select the server.

From the **Configuration** tab, you can view the following Oracle settings:

- ORACLE\_HOME
- ORACLE\_SID
- Oracle User
- Oracle Group
- Oracle Database User
- Oracle Database User Role

## Changing Username / Password for the Oracle Database Account

- Oracle Version
- Oracle Utility



## Changing Username / Password for the Oracle Database Account

After a hierarchy has been created, change the Username and Password using one of the following procedures.

If \$ORACLE\_HOME is on shared (or replicated) storage (common in active-passive configurations):

1. On the system where the Oracle database resource is operational, edit the LifeKeeper configuration file */etc/default/LifeKeeper* and add the following line to the file:

## Changing Username / Password for the Oracle Database Account

```
LK_ORA_NICE=1
```

Do exactly the same on each system in the cluster that has the Oracle resource defined.

2. Use `sqlplus` to change the Oracle user's password.

```
SQL> ALTER USER {username} IDENTIFIED BY {newpassword};
```

3. From the LifeKeeper GUI, right-click on the Oracle Database resource hierarchy, then select **Change Username / Password**.
4. Input **Username** and select **Next**.
5. Input **Password** and select **Next**.
6. Select the **database user role** and click **Apply**. Username and Password will be updated after validating.
7. Select **Done**.
8. Edit the LifeKeeper configuration file on all cluster nodes and make the following change:

```
LK_ORA_NICE=0
```

If `$ORACLE_HOME` is on local storage and each node in the cluster has its own copy of `$ORACLE_HOME` (common in active-active configurations):

1. On the system where the Oracle database resource is operational, edit the LifeKeeper configuration file `/etc/default/LifeKeeper` and add the following line to the file:

```
LK_ORA_NICE=1
```

Do exactly the same on each system in the cluster that has the Oracle resource defined.

2. Use `sqlplus` to change the Oracle user's password.

```
SQL> ALTER USER {username} IDENTIFIED BY {newpassword};
```

3. From the LifeKeeper GUI, right-click on the Oracle Database resource hierarchy, then select **Change Username / Password**.
4. Input user name to use instead temporarily such as `sys` into **Username** and select **Next**.
5. Input **Password** and select **Next**.
6. Select the **database user role** and click **Apply**. Username and Password will be updated after validating.
7. Select **Done**.
8. Put the Oracle database resource "In Service" on one of the backup systems.
9. Once the database is running on the backup system, use `sqlplus` to change the Oracle account password.

## Testing Your Resource Hierarchy

```
SQL> ALTER USER {username} IDENTIFIED BY {newpassword};
```

When making this password change, use the new password that was set in Step 2. This process resets the security tokens in \$ORACLE\_HOME.

10. Put the database “In Service” on each node in the cluster and repeat Step 8.
11. Once the passwords have been changed on all cluster nodes, put the Oracle database back “In Service” on the desired node.
12. From the LifeKeeper GUI, right-click on the Oracle Database resource hierarchy, then select **Change Username / Password**.
13. Input **Username** and select **Next**.
14. Input **Password** and select **Next**.
15. Select the database user role and click **Apply**. Username and Password will be updated after validating.
16. Select **Done**.
17. Edit the LifeKeeper configuration file on all cluster nodes and make the following change:

```
LK_ORA_NICE=0
```

## Testing Your Resource Hierarchy

You can test your Oracle resource hierarchy by initiating a manual switchover. This will simulate a failover of a resource instance from the primary server to the backup server.

### Performing a Manual Switchover from the GUI

You can initiate a manual switchover from the LifeKeeper GUI by selecting Edit, then Resource, then finally In Service from the drop down menu. For example, an in service request executed on a backup server causes the application hierarchy to be placed in service on the backup server and taken out of service on the primary server. At this point, the original backup server is now the primary server and original primary server has now become the backup server.

If you execute the Out of Service request, the application is taken out of service without bringing it in service on the other server.

### Recovery Operations

When the primary server fails, the Oracle Recovery Kit software performs the following tasks:

- Brings Oracle into service on the backup server by bringing in-service the logical interface on one of that server’s physical network interfaces. (Note: This occurs only when there is an IP resource instance defined as a dependency of the Oracle hierarchy.)
- Mounts the file system on the shared disk on that server.

## Recovery Operations

- Starts the daemon processes related to Oracle.

Since session context is lost following recovery, after the recovery, Oracle users must reconnect using exactly the same procedures they used to connect originally.

## Chapter 4: Troubleshooting

The [Message Catalog](#) provides a listing of all messages that may be encountered while using SIOS Protection Suite for Linux and, where appropriate, provides additional explanation of the cause of the errors and necessary action to resolve the error condition. This full listing may be searched for any error code received, or you may go directly to the [Oracle Kit Message Catalog](#) or the [Oracle Listener Message Catalog](#) which contain listings of all messages that may be encountered while utilizing the Oracle Recovery Kit.

- [Oracle Known Issues and Restrictions](#)

### Oracle Known Issues and Restrictions

#### Control File Switchover Failure

If the \$ORACLE\_HOME directory does not recover, the database control files may not be set up properly. For automatic switchover, the control files need to be configured on a shared device during the database creation. If you keep the control files on separate servers, you must manually update both servers when you need to implement changes.

#### Truncated Output

Some versions of Oracle truncate the output when executing the show parameters control\_files in sqldba mode. If your version of Oracle exhibits this behavior, verify the following:

- controlfile parameter. Verify that the controlfile parameter resides in the `$Oracle_HOME/dbs/init#SID.ora` file.
- controlfile devices. Verify that the controlfile devices are on a continuous line, with no new lines, and with each device being separated by a comma.

If an Oracle-related device does not configure properly, then the device can be configured manually using the file system applications available under LifeKeeper Application management.

#### Flash Recovery Destination Located on a Shared Drive

As noted in the configuration section of this document, it is important that the Flash Recovery destination is located on a shared drive. To see where Oracle believes the Flash Recovery Area is, issue the following query (as SYSDBA):

```
SQL> SELECT substr (Name,1,30) Name,  
        (SPACE_LIMIT/1024/1024/1024) Space_Limit_GB,  
        SPACE_USED/1024/1024/1024 Space_Used_GB,
```

## Prevent Failover When Unable to Connect to the Database

```
SPACE_RECLAIMABLE, NUMBER_OF_FILES
FROM V$RECOVERY_FILE_DEST;

NAME                SPACE_LIMIT_GB SPACE_USED_GB SPACE_RECLAIMABLE
-----
NUMBER_OF_FILES
-----
/U01/flash_recovery_area  3.76171875    .156448364      0
                        4
```

Following is an example of how to make a change in `$ORACLE_HOME/dbs/spfile<sid>` to complete this task:

```
SQL> ALTER SYSTEM SET DB_RECOVERY_FILE_DEST='/oracledb/oracle/flash_
recovery_area' scope=both;
System altered.
SQL> show parameter DB_RECOVERY_FILE_DEST;
NAME                TYPE            VALUE
-----
db_recovery_file_dest  string          /oracledb/oracle/flash_recovery_
area
db_recovery_file_dest_size big integer     2G
SQL> commit;
```

## Prevent Failover When Unable to Connect to the Database

When resource health checks are performed, the Oracle ARK checks for running database processes and attempts to connect to the database. To prevent a health check failure caused by reaching the maximum allowed connections, set the following in `/etc/default/LifeKeeper`:

```
LK_ORA_NICE=1
```

**Note:** Setting `LK_ORA_NICE` can mask other types of connection errors caused by a non-functioning database. Use this setting with caution.

## Non-Traditional Location for *oratab*

By default, the Oracle ARK looks for the *oratab* file in `/etc` followed by `/var/opt/oracle`. If the *oratab* file is not located in one of these default locations, then `ORACLE_ORATABLOC` must be set in `/etc/default/LifeKeeper` to the directory containing *oratab*.



## NFS Version 4 Not Supported

The Oracle Recovery Kit supports NFSv3 for shared database storage. NFSv4 is not supported at this time due to NFSv4 file locking mechanisms.

## Creating Oracle Database Hierarchy After Installing Oracle Binaries on Local Storage

If you have elected to install the Oracle binaries (`$ORACLE_BASE`) on local storage on each of your LifeKeeper cluster nodes, you will see a message similar to the following when you create your Oracle database hierarchy in the LifeKeeper GUI.

```
BEGIN create of <SID> on server <server1>
Creating resource instance <SID> on server <server1>
Setting resource state for <SID> on server <server1> to "ISP".

ORACLE_HOME "/opt/oracle/app/oracle/product/11.2.0/dbhome_1" does
not reside on a shared file system. Please be sure that the ORACLE_
HOME directory and associated files are identical on all servers.
Refer to the LifeKeeper Oracle Recovery Kit documentation for more
information.
Creating dependent file system resource "/u00" on <server1>.
Creating dependency between Oracle database "SID (SID)" and the
dependent resource "/u00" on <server1>.
Creating dependency between Oracle database "SID (SID)" and the
listener resource "LSNR.LISTENER" on <server1>.
Performing in-service of new Oracle resource tag=< SID > on
<server1>.
END successful create of <SID> on server <server1>
```

You will also find a similar warning in the LifeKeeper log. If this warning is not heeded and you continue on by extending your hierarchy and then try to bring the database resource in service on another node, you will get a message similar to the following in the LifeKeeper GUI dialog:

```
Put resource "OST" in-service
BEGIN restore of "OST" on server "cae-qa-v39"
Begin the "start [ start.normal ]" of the database "OST" on "cae-qa-
v39".
The "start [ start.normal ]" attempt of the database "OST" appears
to have failed on "cae-qa-v39".
ORA-01078: failure in processing system parameters
LRM-00109: could not open parameter file
'/opt/oracle/app/oracle/product/11.2.0/dbhome_1/dbs/initOST.ora'
Begin the "start [ start.force ]" of the database "OST" on "cae-qa-
v39".
The "start [ start.force ]" attempt of the database "OST" appears to
have failed on "cae-qa-v39".
select 'alter database datafile '||file#||' end backup;' from
v\$backup where status = 'ACTIVE'
```

It is also possible to get a message similar to the following in the dialog box:

```
Put resource "OST" in-service
BEGIN restore of "OST" on server "cae-qa-v39"
Begin the "start [ start.normal ]" of the database "OST" on "cae-qa-
v39".
Initial inspection of "start.normal" failed, verifying failure or
success of received output.
Logon failed with "" for "OST" on "cae-qa-v39". Please check
username/password and privileges.
The "start [ start.normal ]" attempt of the database "OST" appears
to have failed on "cae-qa-v39".
ERROR:

ORA-01031: insufficient privileges

Enter password:

ERROR:

ORA-01005: null password given; logon denied
```

To solve this issue, copy `$ORACLE_BASE/admin` from the primary system where the database instance was created to the backup system (where the hierarchy was extended to) `$ORACLE_BASE/admin`. Also change ownership of this directory to your Oracle username and Oracle group (typically `oracle:oinstall`).

Also copy all `*{$ORACLE_SID}*` (`OST` in this example) files from the primary system in `$ORACLE_BASE/product/11.2.0/dbhome_1/dbs` to `ORACLE_BASE/product/11.2.0/dbhome_1/dbs` on the backup system.

For example, these were the files that were copied from a primary system to the backup, and the ORACLE SID was OST.

```
-rw-r----- 1 oracle oinstall 1544 2012-05-09 11:02 hc_OST.dat
-rw-r----- 1 oracle oinstall 24 2012-01-31 10:22 lkOST
-rw-r----- 1 oracle oinstall 1536 2012-03-05 09:02 orapwOST
-rw-r----- 1 oracle oinstall 2560 2012-05-09 10:58 spfileOST.ora
```

In another example, where the SID was ORA01, the following files were copied:

```
-rw-r----- 1 oracle oinstall 1536 2010-09-08 18:25 orapwORA01
-rw-r----- 1 oracle oinstall 24 2010-09-08 18:25 lkORA01
-rw-r----- 1 oracle oinstall 2560 2010-09-08 18:30 spfileORA01.ora
-rw-r----- 1 oracle oinstall 1544 2010-09-08 18:30 hc_ORA01.dat
```

and a directory

```
peshm_ORA01_0/:
```

## Oracle Listener Stays in Service on Primary Server After Failover

Network failures may result in the listener process remaining active on the primary server after an application failover to the backup server.

### Database Creation Problems

|                 |   |
|-----------------|---|
| <b>Problem:</b> | During DataBase creation using <code>dbca</code> , the following message is received:<br>"ORA-00439 feature not enabled: string"  |
| <b>Action:</b>  | Check the value of the environment variable <code>\$ORACLE_SID</code> . Make sure that is the same as the SID that is being created.  |
| <b>Problem:</b> | During Database creation from scripts, the following message is received:<br>"ORA-01092 ORACLE instance terminated. Disconnection forced"   |
| <b>Action:</b>  | See the alert in the <code>bdump</code> directory. If you see the message "ORA-12714 invalid national character set specified", then check the value of the environment variable <code>\$ORA_NLS33</code> . Make sure that it is set to the correct location.   |
| <b>Problem:</b> | If you encounter problems creating the database from the script generated from <code>dbca</code> , then do the following:   |
| <b>Action:</b>  | <ol style="list-style-type: none"> <li>Be sure to create the following directories if they do not already exist:<br/><br/> <pre>bdump cdump udump &lt;oracle data base directory&gt;/oradata sid &lt;oracle data base directory&gt;/dbs &lt;oracle data base directory&gt;/admin/&lt;SID&gt;</pre> <p>If you need to determine the path to your <code>bdump</code> and <code>udump</code> directories, you can look in the initialization file (<code>init&lt;SID&gt;.ora</code>).</p> </li> <li>Make sure the file <code>\$ORACLE_HOME/dbs/orapw</code> exists; if not, use the <code>orapwd</code> utility to create it.</li> </ol> |

### Database Startup Problems

|                 |  |
|-----------------|--|
| <b>Problem:</b> | During DataBase start-up using <code>sqlplus</code> , the following message is received:<br>"ORA-03113 end-of-file on communication channel"   |
| <b>Action:</b>  | Make sure the initialization file ( <code>init&lt;SID&gt;.ora</code> ) and the password file ( <code>orapw&lt;SID&gt;</code> ) are in the directory <code>\$ORACLE_HOME/dbs</code> . |
| <b>Problem:</b> | During Database startup, the following message is received:<br>"ORA-01092 ORACLE instance terminated. Disconnection forced"  |

|                |   |
|----------------|---|
| <b>Action:</b> | See the alert in the <i>bdump</i> directory. If you see the message "ORA-12701 CREATE DATABASE character set is not known", then check the value of the environment variable <code>\$ORA_NLS33</code> . Make sure that it is set to the correct location. |
|----------------|---|

## inqfail error in the LifeKeeper Log

If an inqfail error appears in your LifeKeeper error log following a failover, you will need to change the `filesystemio` setting.

**Note:** The disk id and server name will be different for each configuration.

To resolve this problem, you will need to change the setting `filesystemio="SETALL"` to `filesystemio="ASYNCH"`

To locate this setting, query the option with the following SQL command:

1. `SQL> show parameter filesystemio;`
2. Use the following commands to change the settings:  

```
SQL> alter system set filesystemio_options=<XXXXXXXX> scope=spfile;
<XXXXXXXX> can be set to
<XXXXXXXX> = {none | setall | direction | asynch}
NONE - no optimization
ASYNCH - enable asynchronous I/O
DIRECTIO - enable direct I/O
SETALL - enables all available features
```

**IMPORTANT:** Oracle needs to be restarted after resetting the parameter.

### Setting Up Oracle to Use Raw I/O

Use the following steps to create an Oracle database that uses shared Raw I/O devices instead of files.

1. Determine the minimum number and sizes of files that you will need to create your database, including control files, tablespaces and redologs. You can create a mixed setup with some of those items as files and others on Raw I/O devices. All of the Raw I/O devices must use shared disk partitions.
2. Create a Raw I/O setup with the necessary number of Raw I/O devices.
  - a. Create the raw devices with the same size or larger than you are going to specify for the Oracle database creation.
  - b. Create raw device mappings in the system initialization file(i.e. `boot.local` or `rc.local`) using the `raw` command. You should add meaningful comments to identify which raw device represents which Oracle file. This is done so that the mapping can be re-established in the case of a re-boot of the system. These mappings should be removed from the file manually once the Raw I/O device is under LifeKeeper protection.
3. Make the raw devices writable for the Oracle database using the following command:

```
chown oracle:dba /dev/raw[0-9]*
```

where the owner and group are specific to your Oracle instance's configuration.
4. Activate the raw device settings by executing the file that contains the mappings.
5. If you already have a database creation script, go directly to Step 6. If not, you may use one of the Oracle Java GUI tools, `dbassist` or `dbca`, to generate your database creation scripts. Using either tool, you must choose to **"Save As Script"**. Do not choose to create the database.

#### Notes:

- In `dbca`, the **"New Database"** template must be selected to generate scripts. Change filenames to shared devices and adjust the values for your configuration if necessary.
  - The DB creation process should not be started at this point! The `dbassist` tool checks to see if the file specified for each tablespace already exists and will not proceed if it does. The `dbca` tool prompts to confirm that it will overwrite the files but fails on raw devices. In either case, you are unable to use raw devices directly from these tools.
6. The database creation scripts (either the existing ones or those created by `dbassist` or `dbca`) must be edited. The desired filename (including the path) must be replaced with the full path name of the Raw I/O device. The affected files should include (at minimum) the file's database creation file (for the

## Setting Up Oracle to Use Raw I/O

CREATE DATABASE command) and tablespace creation file (for the CREATE TABLESPACE command). Depending on what options you selected in `dbassist` or `dbca`, there may more files to be edited. Also, edit the initialization file to change the control files to Raw I/O devices, if desired. The initialization file is located in the directory with the creation script. The result looks like this for the data file:

```
...
CREATE DATABASE "LK"
    maxdatafiles 254
    maxinstances 8
    maxlogfiles 32
    character set US7ASCII
    national character set US7ASCII
DATAFILE '/dev/raw/raw1' SIZE 260M AUTOEXTEND ON NEXT 10240K
logfile '/ora/LK/redo01.log' SIZE 500K,
        '/ora/LK/redo02.log' SIZE 500K,
        '/ora/LK/redo03.log' SIZE 500K;
...
```

The Raw I/O device must be the minimum size required by Oracle for the data that will be stored.

7. Now create the database by running the script that you created in step #5.
8. Be sure to check the create log for any database or tablespace errors that may have occurred.
9. If you have trouble creating the database with the creation scripts, or you want to add tablespaces on raw devices later, you must create the database with the applicable tool (i.e. `dbassist` or `dbca`). Then, add the Raw I/O device data files by executing a command similar to the following from the `sql` utility:

```
tablespace RAWTS
DATAFILE '/dev/raw/raw217' SIZE 50M REUSE
DEFAULT STORAGE (INITIAL 50K NEXT 50K
MINEXTENTS 1 MAXEXTENTS 4) ONLINE
```

10. Add `udev` rules to make the raw device permissions and ownership persistent across reboots, switchovers and failovers.

In the Linux 2.6 kernel, the `udev` system is the default method through which the kernel controls the creation of the special files such as raw devices. When used by Oracle, raw devices require specific ownership and permission settings. These specific settings conflict with the kernel default settings. Addressing the specific settings requires the use of `udev` rules to set the ownership of the raw device to the Oracle user and Oracle group used with the LifeKeeper protected Oracle SID.

**Note:** In some OS distributions, rules for creating devices and rules for setting device

permissions must be separate. Check your OS distribution udev documentation for more information.

The following are example udev rules that may work for your OS distribution:

```
KERNEL=="raw10", RUN+="/bin/chown oracle:oinstall
/dev/raw/raw10"

KERNEL=="raw[3-5]*", OWNER="oracle", GROUP="oinstall",
MODE="660"
```

The udev rules created must be applied to all nodes in the LifeKeeper cluster prior to bringing the resource hierarchy into service.

## Adding a Tablespace After Creating Hierarchy

If a tablespace is added on a Raw I/O device after the Oracle hierarchy has been created in LifeKeeper, you must create a LifeKeeper Raw I/O hierarchy via the GUI and manually create a dependency between the Oracle resource (as parent) and the Raw I/O resource (as child).

## Creating an Oracle Listener for Multiple Resources

You may want to create an Oracle Listener if any of the following statements are true for your system configuration:

- Multiple Listeners are defined for Multiple Oracle SIDs
- The Oracle Listener is a critical component in your configuration
- A Single Listener is defined for Multiple Oracle SIDs



**NOTE:** If multiple listeners are intended to be protected with SPS, the collection of LISTENER/SID\_LIST\_LISTENER stanzas must all be unique. (For example, the 10g installation's *listener.ora* file would contain the stanzas LISTENER\_10G\_1/SID\_LIST\_LISTENER\_10G\_1 and the 11g installation's *listener.ora* file would contain the stanzas LISTENER\_11G\_1/SID\_LIST\_LISTENER\_11G\_1.)

This process will allow protection of listener(s) within LifeKeeper to accommodate various listener(s) and SIDs combinations.

If you are creating a Listener for multiple resources, follow these procedures.

1. From the LifeKeeper GUI menu, select **Edit**, then **Server**. From the drop-down menu, select **Create Resource Hierarchy**.

**IMPORTANT:** The Oracle Application must be running when you create the resource

2. A dialog box will appear with a drop down list box with all recognized Recovery Kits installed within the cluster. Select **Oracle Database Listener** from the drop down listing. Click **Next** to proceed to the next dialog box.

- You will be prompted to enter the following information. When the **Back** button is active in any of the dialog boxes, you can go back to the previous dialog box. This is especially helpful should you encounter an error that might require you to correct previously entered information. If you click **Cancel** at any time during the sequence of creating your hierarchy, LifeKeeper will cancel the entire creation process.

| Field                                   | Tips  |
|---|---|
| <b>Switchback Type</b>                  | Choose either intelligent or automatic. This dictates how the Listener resource will be switched back to this server when the server comes back up after a failover. If using data replication, choose intelligent as the switchback type.<br><br><b>Note:</b> The switchback type must match that of the dependent resources (IP and volume resources) used by the Listener resource, or else the create will fail.  |
| <b>Server</b>                           | Select the server on which you want to create the hierarchy.  |
| <b>Listener Configuration File Path</b> | Select the full path to the Oracle listener configuration file.   |
| <b>Listener Names(s)</b>                | Select the name(s) of the Oracle Listener(s) to provide protection for with this resource instance.   |
| <b>Listener Executable(s)</b>           | Select the path to the Oracle listener executable. The listener executable is required to start, stop, monitor and recover the specified Oracle listener(s).  |
| <b>Listener Protection Level</b>        | Select one of the following levels:<br><br>Full Control (Start, Stop, Monitor and Recover)<br><br>Intermediate Control (Start, Monitor and Recover)<br><br>Minimal Control (Start and Monitor Only)   |
| <b>Listener Recovery Level</b>          | Select the level of recovery for the specified listener:<br><br>Standard (On) – Enable standard LifeKeeper recovery. If Standard (On) is selected, all listener failures will be tried locally, and if necessary, trigger a failover to an available backup server.<br><br>Optional (off) – Enable optional LifeKeeper recovery. If Optional (Off) is selected, all listener failures will be tried locally, but will not cause a failover to an available backup server. |
| <b>IP Address Name(s)</b>               | Select the IP Address resource name that will be protected as dependents of this resource hierarchy. IP Address associated with the selected listener(s) are displayed in the choice list. Select None if no IP resources are required for this configuration,  |
| <b>Listener Tag</b>                     | Enter a unique name for the resource on the server. The valid characters allowed for the name are letters, digits, and the following special characters: - _ . /  |

- Select the **Create** button to start the hierarchy creation. An information box appears and LifeKeeper will validate that you have provided valid data to create your database listener resource hierarchy. If



LifeKeeper detects a problem, an ERROR will appear in the information box. If the validation is successful, your resource will be created.


- The **Pre-Extend Wizard** dialog will appear stating that you have successfully created the resource hierarchy and you will be prompted to select the following information. If you are unfamiliar with the **Extend** operation, click **Next** after making a selection in each dialog box. If you are familiar with the **LifeKeeper Extend Resource Hierarchy** defaults and want to bypass the prompts for input/confirmation, click **Accept Defaults**.

| Field                    | Tips   |
|--------------------------|--|
| <b>Target Server</b>     | Select a Target Server to which the hierarchy will be extended. If you select Cancel before extending the resource hierarchy to at least one other server, LifeKeeper will provide no protection for the applications in the hierarchy.  |
| <b>Switchback Type</b>   | This dictates how the Oracle Listener instance will be switched back to this server when it comes back into service after a failover to the backup server. You can choose either intelligent or automatic. The switchback type can be changed later, if desired, from the General tab of the Resource Properties dialog box.<br>Note: Remember that the switchback strategy must match that of the dependent resources to be used by the Oracle Listener resource. |
| <b>Template Priority</b> | This field appears only if you did NOT extend directly from the Create function.) Enter a number between 1 and 999 to specify the template server's priority in the cascading failover sequence for this resource. A lower number means a higher priority. LifeKeeper assigns the number "1" to the server on which the hierarchy was created. No two servers can have the same priority for a given resource.   |
| <b>Target Priority</b>   | Enter a number between 1 and 999 to specify the target server's priority in the cascading failover sequence for this resource. A lower number means a higher priority. LifeKeeper offers a default of 10 for the first server to which a hierarchy is extended.  |

- After receiving the message that the pre-extend checks were successful, click **Next** and enter the following information.

| Field                            | Tips  |
|----------------------------------|---|
| Listener Configuration File Path | Select the full path to the Oracle Listener configuration file.   |
| Listener Executable (s) Path     | Select the path to the Oracle Listener executables. The listener executables are required to start, stop, monitor and recover the specified Oracle listener(s).   |
| Listener Tag                     | This field is automatically populated with a unique name for the new Oracle Listener resource instance on the primary server. The default naming pattern will be displayed for you. You may type in another unique name. The valid characters allowed for the Listener tag are letters, digits and the following special characters - : . / |


7. Click **Extend**. The **Hierarchy Integrity Verification** window displays with the following message, **Hierarchy Verification Finished**. Click **Next Server** or **Finish**.

1. Select a **resource** and then the  button from the **Resource** toolbar to update the protection level of the resource.

2. Enter the following information.

| Field                            | Tips   |
|----------------------------------|--|
| <b>Listener Protection Level</b> | Select one of the following:<br>Full Control (Start, Stop, Monitor and Recover)<br>Intermediate Control (Start, Monitor and Recover)<br>Minimal Control (Start and Monitor Only) |

3. Click **Update** to change the **Protection Level** from the current state to the new state. Select **Cancel** to leave the value unchanged.


1. Select a **listener** and then the  button from the **Resource** toolbar to update the recovery level of the resource.

2. Enter the following information.

| Field                          | Tips  |
|--------------------------------|---|
| <b>Listener Recovery Level</b> | Select the level of recovery for the specified listener(s).<br><br>Standard (On_ – enables a standard LifeKeeper recovery. If Standard, (On) is selected, all listener failures will be tried locally and if necessary trigger a fail over to an available backup server.<br><br>Optional (Off) – enables optional LifeKeeper recovery. If Optional (Off) is selected, all listener failures will be tried locally, but will not cause a fail over to an available backup server. |
| <b>Update Confirmation</b>     | Select the Update button to change the Recovery Level from the current state to the new state.<br><br>Select Cancel to leave the current value unchanged.   |

3. Click **Update** to change the Recovery Level from the current state to the new state. Select **Cancel** to leave the current value unchanged.

## Updating the Protected Listener(s)

1. Select a **listener** and then the  button from the **Resource** toolbar to update your protected listener (s).
2. Enter the following information.

| Field                     | Tips   |
|---------------------------|--|
| <b>Listener Name (s)</b>  | Select the name or names of the Oracle Listener(s) to provide protection for with this resource instance.  |
| <b>IP Address Name(s)</b> | Select the IP Address resource name(s) that will be added as a dependent of this resource hierarchy.<br><br>Select None if no additional IP Resources are required for this configuration. |

3. Click **Update** to change the **Protected Listener(s)** and **IP assignment** from the current state to the new state. Select **Cancel** to leave the current value unchanged.

If you select **Update**, a dialog displays stating that the Protected Listeners for the specific resource is being updated. Click **Finish**.