Installing and Configuring Oracle9*i* RAC with Sistina GFS™

Corporate Headquarters
Sistina Software, Inc.
720 Washington Ave. SE, Suite 200
Minneapolis, MN 55414
USA
http://www.sistina.com
Tel: 612-638-0500
Fax: 612-638-0525

Document Number:TD4003A

Contents

Index

Preface

This book provides information about installing and configuring Oracle9*i* Real Application Clusters (RAC) for use with Sistina GFS™. The document contains procedures for a sample network that can be used as a reference for your network.

Audience

This book is intended primarily for Oracle database administrators and Linux system administrators who are familiar with the following activities:

- Oracle database administration procedures
- Linux system administration procedures, including kernel configuration
- Installing and configuring shared-storage networks, such as Fibre Channel SANs

Related Documentation

For additional information, refer to the following documentation:

- Sistina *GFS Administrator's Guide Release 5.2.1*
- Sistina GFS product release notes
- Relevant Oracle9*i* RAC documentation

Organization

This book contains the following chapters.

Chapter	Description
Chapter 1 - Overview	Describes sample network, requirements, and prerequisites.
Chapter 2 - Installing and Configuring GFS	Describes GFS installation procedures that are specific to an Oracle9*i* RAC installation.
Chapter 3 - Installing and Configuring Oracle9i RAC	Describes procedures for installing Oracle9*i* RAC that are required for using GFS.

Conventions

This book uses the following conventions.

Command Usage

boldface font	Commands, keywords, programs, subsystem components, file names, file paths, directories, literal elements in configuration files and file sections, and other item names that are defined by a user or predefined by an operating system. All terms in boldface font are typed literally.
italic font	In command usage or description, arguments and variables for which you supply values.
`screen` font	Examples of text similar to what may be displayed or examples of files similar to what you may create.
`italic screen` font	In example text, arguments for which you supply values, or output text that is variable.
`boldface screen` font	In example text, text that is typed literally.
→	Used with example text for emphasis.

Notes

Note Means *take note*. A note provides suggestions or references to additional information and material that may not be available in this book.

Sistina Customer Support

If you have a support contract with a support organization besides Sistina, please contact that organization.

If you have a support contract with Sistina, you can contact Sistina support directly as follows:

- E-mail—Please send a question or a description of your problem to support@sistina.com.
- Phone—Support personnel can be reached at 1-866-SISTINA (1-866-747-8462) during the hours of 07:00-19:00 central time.
- Web—Online support is available at at the Sistina Support Web page, https://secure.sistina.com/pub/support/.

If you are a new customer or do not have a support contract, you can obtain more information about a support contract as follows:

- E-mail—Please send inquiries to info@sistina.com.
- Phone—Call 612-638-0500.
- Web—Information about support is available at http://www.sistina.com/services_custsupport.html.

Chapter 1 - Overview

This book provides information about installing and configuring Oracle9*i* Real Application Clusters (RAC) Release 2 on a Linux cluster that uses Sistina GFS™. It contains procedures for a sample network that can be used as a reference for your network. You should use this book in conjunction with the appropriate guides for installing and maintaining Oracle9*i* RAC on a Linux cluster, and the appropriate GFS administrator's guide.

This chapter provides an introduction to the sample network used in the book, requirements based on the sample network, and prerequisites for installing and configuring Oracle9*i* RAC on a Linux cluster that uses GFS. The chapter consists of the following sections:

Sample Network

The sample network used in this book consists of the following components (see Figure 1):

- WTI network power switch (NPS)
- Four-node Linux cluster using Sistina GFS™
- Brocade switch
- RAID-5 array

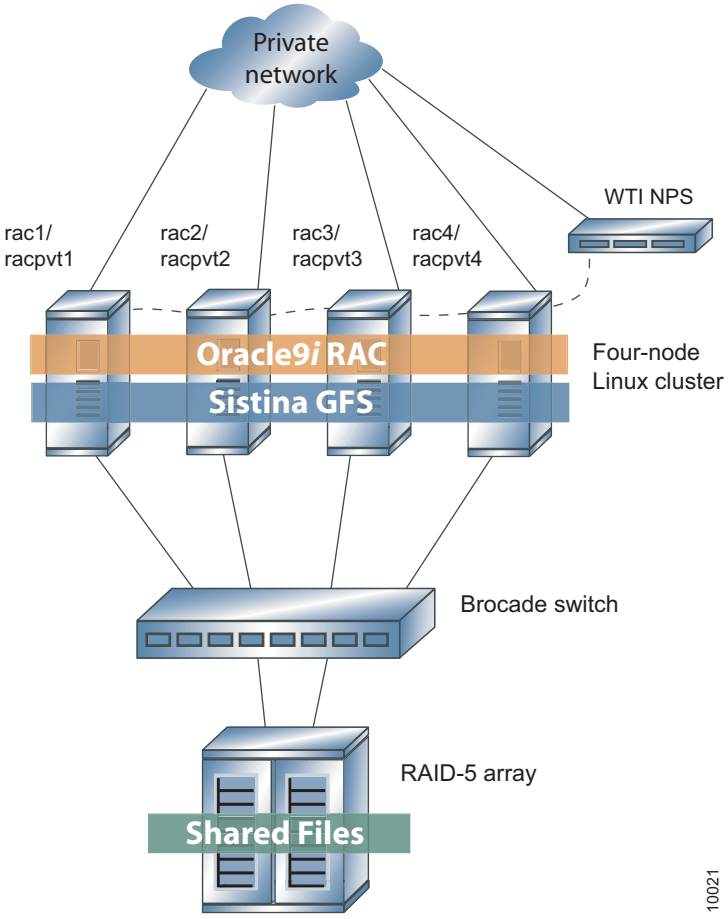

Figure 1 Sample Network

The cluster is attached to the RAID-5 array via the Brocade switch. Also, the cluster is connected to the WTI network power switch (NPS) for fencing. Table 1 lists the sample-network node names and IP addresses for the public network and the private network.

Table 1 Sample-Network Node Names and IP Addresses

Public Network		Private Network	
Node Name	**IP Address**	**Node Name**	**IP Address**
rac1	10.3.0.101	racpvt1	192.168.100.101
rac2	10.3.0.102	racpvt2	192.168.100.102
rac3	10.3.0.103	racpvt3	192.168.100.103
rac4	10.3.0.104	racpvt4	192.168.100.104

Note The procedures in this book are based on the assumption that you will be using a private network for your Oracle cluster software, and that the Oracle management node is rac1/racpvt1.

Requirements

This section provides requirements for the sample network used in this book and consists of hardware requirements (Table 2), software requirements (Table 3), and sample disk layout information (Table 4).

Table 2 Hardware Requirements

Requirement	Description
Memory	A minimum of 512 MB of RAM on each node, more if Java VM is used
Swap space	Disk Space equal to system's physical memory, or 1 GB, whichever is greater
Disk space	6 GB to 10 GB
Servers	With dual NICs, a Fibre Channel HBA, external SCSI, or a gigabit Ethernet card

Table 3 Software Requirements

Requirement	Description
Operating system	Red Hat Enterprise Server 2.1 or SuSE Enterprise Server 8, and appropriate patches
DBMS	Oracle9*i* R2 and 9.2.0.3 patch
Cluster file system	GFS Release 5.2 or later

Table 4 Sample Disk Layout

Mount	Purpose
/mnt/oracle	ORACLE_BASE, ORACLE_HOME
/mnt/oradata	Data files
/mnt/oraindex	Index files
/mnt/oraundo	Undo/redo log files
/oraraw1, /oraraw2	Each with 100 MB size raw partitions (quorum pools)

Note Refer to the GFS administrator's guide for setting up the file systems, configuration files, and for required processes (for example, **ccsd** and **lock_gulmd**).

Prerequisites

Ensure that your servers can connect to each other. Your /**etc/hosts** file should look similar to the one in Example 1.

Example 1 Sample File: **/etc/hosts**

```
#
# hosts      This file describes a number of hostname-to-address
#               mappings for the TCP/IP subsystem.  It is mostly
#               used at boot time, when no name servers are running.
#               On small systems, this file can be used instead of a
#               "named" name server.
# Syntax:
#
# IP-Address  Full-Qualified-Hostname  Short-Hostname
#

10.3.0.101      rac1.hq.sistina.com     rac1
10.3.0.102      rac2.hq.sistina.com     rac2
10.3.0.103      rac3.hq.sistina.com     rac3
10.3.0.104      rac4.hq.sistina.com     rac4
192.168.100.101 racpvt1.hq.sistina.com racpvt1
192.168.100.102 racpvt2.hq.sistina.com racpvt2
192.168.100.103 racpvt3.hq.sistina.com racpvt3
192.168.100.104 racpvt4.hq.sistina.com racpvt4
```

If your private network is secure, you may want to allow for unrestricted access to the other nodes. You may also want to set up unrestricted access to the nodes through the private network.

The sample configuration used in this book uses a RAID array in a RAID-5 configuration. If possible, you would want to use an array that allows multiple LUNs to be exported. If your array only supports one LUN, then partition the array according to Table 5 on page 6.

Table 5 Partitions If Using One LUN

Partition	Quantity	Size
GFS cluster archive	1	100 MB
Oracle Quorum disk	1	100 MB
Oracle cluster configuration data	1	100 MB
Oracle binaries, logs, trace files, etc.	1	10 GB
Oracle data files	1	20 GB (minimum)
Oracle index files	1	10 GB (minimum)
Oracle undo tablespaces (and redo logs)	1	10 GB (minimum)

Chapter 2 - Installing and Configuring GFS

This chapter describes installing and configuring GFS for use with Oracle9*i* RAC using the sample network described in "Chapter 1 - Overview." This chapter consists of the following sections:

Installation and Configuration Procedure

Installing and configuring GFS consists of the following steps:

Step 1 Installing GFS.

Step 2 Configuring pools.

Step 3 Creating GFS configuration files.

Step 4 Starting the lock server and loading the **hangcheck-timer** module.

Step 5 Creating and mounting GFS file systems.

Installing GFS

To install GFS, follow these steps:

Step 1 Follow the installation instructions for installing the GFS RPMs that were received when downloading them.

Step 2 Follow the directions in the Sistina GFS administrator's guide for preparing and setting up the server to use GFS. In particular, make sure that any additional packages are installed that may not have come with the base Linux distribution being used (for example, Perl::Net-Telnet).

Step 3 Proceed to the next section, "Configuring Pools."

Configuring Pools

Lay out the GFS volumes (pools) that will be used for GFS file systems. For reliability, striping across multiple RAID arrays is preferred to single drives. Example layouts are provided in this procedure, but should be adapted to the hardware environment on which GFS is being deployed. The procedures provide instructions on configuring a minimum configuration.

To configure the pools, follow these steps:

Step 1 Create two 100-MB pools for Oracle's cluster and server control software (this supports the cluster manager and is not used for the database). Suggested names for the pools are **oraraw1** and **oraraw2**, with corresponding pool configuration files, **oraraw1.pool** and **oraraw2.pool**. Create and save **oraraw1.pool** and **oraraw2.pool** using the following as examples:

oraraw1.pool

```
poolname oraraw1
subpools 1
subpool 0 0 1 gfs_data
pooldevice 0 0 /dev/sda6
```

oraraw2.pool

```
poolname oraraw2
subpools 1
subpool 0 0 1 gfs_data
pooldevice 0 0 /dev/sda7
```

Step 2 Create a small (for example, 8 MB) pool for the GFS CCS archive. The example uses a 100-MB partition. The suggested name for the pool is **ccs_archive**, with corresponding pool configuration file, **ccs_archive.pool**. Create and save **ccs_archive.pool** using the following as an example:

```
poolname ccs_archive
subpools 1
subpool 0 0 1 gfs_data
pooldevice 0 0 /dev/sda5
```

Step 3 Create a pool for the **$ORACLE_HOME** directory. It should be at least 10 GB. The following example uses a 20-GB partition. The suggested name for the pool is **oracle_base**, with corresponding pool configuration file, **oracle_base.pool**. Create and save **oracle_base.pool** using the following as an example:

```
poolname oracle_base
subpools 1
subpool 0 0 1 gfs_data
pooldevice 0 0 /dev/sda1
```

Step 4 Create a pool that is larger than 10 GB for each of the undo and index tablespace datafiles. The following examples each use a 98-GB partition. Suggested names for the pools are **oraundo** and **oraindex**, with corresponding pool configuration files, **oraundo.pool** and **oraindex.pool**. Create and save **oraundo.pool** and **oraindex.pool** using the following as examples:

oraundo.pool

```
poolname oraundo
subpools 1
subpool 0 0 1 gfs_data
pooldevice 0 0 /dev/sda3
```

oraindex.pool

```
poolname oraindex
subpools 1
subpool 0 0 1 gfs_data
pooldevice 0 0 /dev/sda8
```

Step 5 Create a pool that is larger than 20 GB for Oracle system and data files. The example uses a 200-MB partition. The suggested name for the pool is **oradata**, with corresponding pool configuration file, **oradata.pool**. Create and save **oradata.pool** using the following as an example:

```
poolname oradata
subpools 1
subpool 0 0 1 gfs_data
pooldevice 0 0 /dev/sda2
```

Step 6 Create the required GFS pools using the GFS **pool_tool** command. This need only be done from one node of the cluster. See the following example:

```
rac1 # pool_tool -c oraraw1.pool
Pool label written successfully from oraraw1.pool.
rac1 # pool_tool -c oraraw2.pool
Pool label written successfully from oraraw2.pool.
rac1 # pool_tool -c ccs_archive.pool
Pool label written successfully from ccs_archive.pool.
rac1 # pool_tool -c oracle_base.pool
Pool label written successfully from oracle_base.pool.
rac1 # pool_tool -c oraundo.pool
Pool label written successfully from oraundo.pool.
rac1 # pool_tool -c oraindex.pool
Pool label written successfully from oraindex.pool.
rac1 # pool_tool -c oradata.pool
Pool label written successfully from oradata.pool.
```

Step 7 Allow your systems to see the pools that were created in the previous step by running the **pool_assemble** command on all nodes. See the following example:

```
rac1 # pool_assemble
ccs_archive assembled.
oracle_base assembled.
oradata assembled.
oraindex assembled.
oraraw1 assembled.
oraraw2 assembled.
oraundo assembled.

rac2 # pool_assemble
⋮
rac3 # pool_assemble
⋮
rac4 # pool_assemble
⋮
```

Step 8 Proceed to the next section, “Creating GFS Configuration Files.”

Creating GFS Configuration Files

Creating the GFS configuration files consists of creating and storing the configuration files for the GFS Cluster Configuration System (CCS), and starting the daemon for the GFS CCS. This section describes how to create the GFS configuration files for the sample network described in “Chapter 1 - Overview.” The files to be created and stored are **license.ccs**, **cluster.ccs**, **fence.ccs**, and **nodes.ccs**.

Note When saving the files, use the file name extension, **.ccs**. Keywords, case, and quotation marks are significant in the GFS configuration files. Please refer to the Sistina GFS administrator’s guide for syntax rules.

This configuration has the following key characteristics:

- Number of GFS nodes—4, nodes **rac1**, **rac2**, **rac3**, and **rac4**.
- Number of lock server nodes—3. The lock servers will be run on nodes **rac1**, **rac2**, and **rac3**.
- Locking protocol—RLM, using LOCK_GULM.

- Fencing device—WTI NPS.
- Cluster name—**gfsrac**.

To create the GFS configuration files, follow these steps:

Step 1 Set up a temporary directory to save the GFS configuration files.

Step 2 Obtain a Sistina license file and save it to the temporary directory as **license.ccs**.

Step 3 Create the following **cluster.ccs** file and save it to the temporary directory:

```
#### cluster.ccs ###################################
cluster {
   name = "gfsrac"
   lock_gulm {
         servers = [ "rac1","rac2","rac3" ]
   }
}
```

Step 4 Create the following **fence.ccs** file and save it to the temporary directory:

```
### fence.ccs ######################################
fence_devices {
   nps {
      agent = "fence_wti"
      ipaddr = "10.3.0.150"
      login = "nps"
      passwd = "password"
   }
}
```

Step 5 Create the following **nodes.ccs** file and save it to the temporary directory:

```
#### nodes.ccs ########################################
nodes {
   rac1 {
      ip_interfaces {
         eth0 = "10.3.0.101"
      }
      fence {
         power {
            nps {
               port = 1
            }
         }
      }
   }
   rac2 {
      ip_interfaces {
         eth0 = "10.3.0.102"
      }
      fence {
         power {
            nps {
               port = 2
            }
         }
      }
   }
   rac3 {
      ip_interfaces {
         eth0 = "10.3.0.103"
      }
      fence {
         power {
            nps {
               port = 3
            }
         }
      }
   }
   rac4 {
      ip_interfaces {
         eth0 = "10.3.0.104"
      }
      fence {
         power {
            nps {
               port = 4
            }
         }
      }
   }
}
```

Step 6 Using the **ccs_tool create** command, write the configuration information from the temporary directory (the *Directory* variable) to the GFS cluster archive pool as follows:

```
rac1# ccs_tool create Directory /dev/pool/ccs_archive
```

Step 7 Start the CCS daemon, **ccsd**, on all nodes as follows:

```
rac1 # ccsd -d /dev/pool/ccs_archive

rac2 # ccsd -d /dev/pool/ccs_archive

rac3 # ccsd -d /dev/pool/ccs_archive

rac4 # ccsd -d /dev/pool/ccs_archive
```

Step 8 Proceed to the next section, "Starting the Lock Server and Loading hangcheck-timer."

Starting the Lock Server and Loading hangcheck-timer

To start the lock servers and load **hangcheck-timer**, follow these steps:

Step 1 Start the lock servers on **rac1**, **rac2**, and **rac3** as follows:

```
rac1 # lock_gulmd

rac2 # lock_gulmd

rac3 # lock_gulmd
```

Step 2 If the **hangcheck-timer** module is not yet loaded, load it on each node as follows:

```
rac1# insmod hangcheck-timer hangcheck_tick=300 hangcheck_margin=1800

rac2# insmod hangcheck-timer hangcheck_tick=300 hangcheck_margin=1800

rac3# insmod hangcheck-timer hangcheck_tick=300 hangcheck_margin=1800

rac4# insmod hangcheck-timer hangcheck_tick=300 hangcheck_margin=1800
```

Step 3 Proceed to the next section, "Creating and Mounting GFS File Systems."

Creating and Mounting GFS File Systems

The procedure in this section creates GFS file system(s) using GFS pools **oraundo**, **oradata**, and **oracle_base**. All file systems that will contain Oracle data files should be created with a blocksize of 512. Oracle9*i* RAC requires this because it assumes it will be running on raw devices where the block size is 512 bytes by default. This is an Oracle limitation.

To create and mount GFS file systems, follow these steps:

Step 1 Create GFS file systems as follows.

Note The file systems need only be created from one node in the cluster.

```
rac1 # gfs_mkfs -p lock_gulm -t gfsrac:oraundo -j 4 /dev/pool/oraundo -b 512
rac1 # gfs_mkfs -p lock_gulm -t gfsrac:oraindex -j 4 /dev/pool/oraindex -b 512
rac1 # gfs_mkfs -p lock_gulm -t gfsrac:oradata -j 4 /dev/pool/oradata -b 512
rac1 # gfs_mkfs -p lock_gulm -t gfsrac:oracle_base -j 4 /dev/pool/oracle_base
```

Step 2 Make mount points for each of the file system you created. This needs to be done on every node that will mount file systems. The following example shows making mount points on node **rac1**:

```
rac1 # cd /mnt
rac1 # mkdir oracle_base oradata oraundo oraindex
```

Step 3 Link the raw devices to the Oracle quorum pools. The following example shows linking the raw devices to the Oracle quorum pools on node **rac1**:

```
rac1 # /usr/sbin/raw /dev/raw/raw1 /dev/pool/oraraw1
rac1 # /usr/sbin/raw /dev/raw/raw2 /dev/pool/oraraw2
```

Step 4 Mount the GFS file systems. Perform this step on every node in the cluster that will access GFS file systems. The following example shows mounting the GFS file systems on node **rac1**:

```
rac1 # mount -t gfs /dev/pool/oracle_base /mnt/oracle_base
rac1 # mount -t gfs /dev/pool/oradata /mnt/oradata
rac1 # mount -t gfs /dev/pool/oraundo /mnt/oraundo
rac1 # mount -t gfs /dev/pool/oraindex /mnt/oraindex
```

Step 5 If **/var/opt/oracle** does not exist, create it and allow Oracle to use it.

Step 6 Proceed to the next chapter,"Chapter 3 - Installing and Configuring Oracle9i RAC."

Chapter 3 - Installing and Configuring Oracle9*i* RAC

This chapter describes how to install and configure Oracle9*i* RAC for use with GFS, using the sample network described in "Chapter 1 - Overview." This chapter consists of the following sections:

Installation and Configuration Procedure

Installing and configuring Oracle 9i RAC consists of the following steps:

Step 1 Preparing the Oracle nodes.

Step 2 Installing and configuring Oracle.

Step 3 Creating an Oracle database.

Preparing Oracle Nodes

To prepare the Oracle nodes, follow these steps:

Step 1 At each node, create the Oracle group. For example:

```
# groupadd dba -g 501
```

Step 2 Make the Oracle user's home directory. For example:

```
# mkdir -p /home/oracle
```

Step 3 At each node, create the Oracle user. Make sure that the Oracle user primary group is the **dba** group. For example:

```
# useradd -c "Oracle Software Owner" -G dba -u 101 -m -d /home/oracle -s /bin/bash oracle
```

Step 4 Grant unlimited **ulimit** to **oracle**.

Step 5 Make sure that **oracle:dba** owns the following mount points and quorum pool devices:

- **/dev/raw/raw1**
- **/dev/raw/raw2**
- **/mnt/oracle_base**
- **/mnt/oradata**
- **/mnt/oraundo**
- **/mnt/oraindex**

Step 6 Verify and adjust kernel parameters.

a. Verify that operating system kernel parameters are set to the following values:

Kernel Parameter	Setting	Purpose
SHMMAX	2147483648	Maximum allowable size of one shared memory segment.
SHMMIN	1	Minimum allowable size of a single shared memory segment.
SHMMNI	100	Maximum number of shared memory segments in the entire system.
SHMSEG	10	Maximum number of shared memory segments one process can attach.
SEMMNI	100	Maximum number of semaphore sets in the entire system.
SEMMSL	100	Minimum recommended value. SEMMSL should be 10 plus the largest PROCESSES parameter of any Oracle database on the system.
SEMMNS	32000	Maximum semaphores on the system. This setting is a minimum recommended value. SEMMNS should be set to the sum of the PROCESSES parameter for each Oracle database, add the largest one twice, plus add an additional 10 for each database.
SEMOPM	100	Maximum number of operations per semop call.

b. Make sure that those kernel parameters are set at startup by including the following commands in the startup script of each node:

```
# export SEMMSL=100
# export SEMMNS=32000
# export SEMOPM=100
# export SEMMNI=100
# echo  $SEMMSL $SEMMNS $SEMOPM $ SEMMNI > /proc/sys/kernel/sem
# export SHMMAX=2147483648
# echo $SHMMAX > /proc/sys/kernel/shmmax
```

c. To verify the settings, use the following commands:

```
# cat /proc/sys/kernel/sem
# cat /proc/sys/kernel/shmmax
```

d. To to increase the maximum number of file handles, use **/etc/sysctl.conf** or include this command in the startup script of each node:

```
# echo 65536 > /proc/sys/fs/file-max
```

Step 7 Establish Oracle environment variables.

a. Set the following Oracle environment variables.

Note So that you do not have to set the environment values each time you log in, you can save the values in a **.login** or **.profile** file.

Environment Variable	Suggested Value
ORACLE_HOME	For example: **/mnt/oracle/product/920**
ORACLE_TERM	**Xterm**
ORA_NLS33	**$ORACLE_HOME/ocommon/nls/admin/data**
PATH	**/mnt/oracle/product/9.2.0/bin: /usr/ccs/bin:/usr/bin/X11/:/usr/local/bin** and any other items you require in your PATH
DISPLAY	*ip-address***:0.0**
TMPDIR	Set a temporary directory path for TMPDIR with at least 100 MB of free space to which the OUI has write permission.
ORACLE_SID	Set this to what you will call your database instance. This should be a *unique* value on each node.
umask	022

b. Create the directory **/var/opt/oracle** and set ownership to the **oracle** user if it does not exist. For example:

```
$ mkdir /var/opt/oracle
$ chown oracle.dba /var/opt/oracle
```

Step 8 Proceed to the next section, "Installing and Configuring Oracle Components."

Installing and Configuring Oracle Components

Installing and configuring Oracle components consists of the following tasks:

Step 1 Installing Cluster Manager, page 21

Step 2 Configuring Cluster Manager, page 23

Step 3 Installing Oracle RDBMS, page 26

Step 4 Applying Patch, page 27

Installing Cluster Manager

Installing Cluster Manager includes using the Sistina CDPN (Context Dependent Path Name) for the following directories:

- **$ORACLE_HOME/network**
- **$ORACLE_HOME/oracm**
- Database's archive directory

To install the Cluster Manager, follow these steps:

Step 1 At the Oracle management node, log in to the Oracle account and start an X11 windows session if it does not start automatically.

Note This step is performed only from one node, the Oracle management node.

Step 2 Run the Oracle Universal Installer from a CD-ROM or from stage directories.

a. From CD-ROM—Change directory and run installer. For example:

```
$ cd /home/oracle
$ /cdrom/runInstaller
```

b. From stage directories—Change directory and run installer. For example:

```
$ cd /stage/Disk1
$ ./runInstaller
```

Step 3 Running the Oracle Universal Installer causes a **Welcome** dialog box to be displayed. Click **Next**. Clicking **Next** causes the **Inventory Location** dialog box to be displayed.

Step 4 At **What would you like as the base directory?**, verify that the base directory is set to the **ORACLE_BASE** directory from your environment variable. If so, click **OK**. Clicking **OK** causes the **UNIX Group Name** dialog box to be displayed.

Step 5 At the **UNIX Group Name** dialog box, enter the group name (created in Step 1 of "Preparing Oracle Nodes" on page 18) and click **Next**. Clicking **Next** may cause an **Oracle Universal Installer** dialog box to be displayed that prompts you to run **/tmp/orainstRoot.sh**. In another terminal window, log in as **root**, and run the command as stated in the dialog box. After running the command, click **Continue** in the **Oracle Universal Installer** dialog box. Clicking **Continue** causes the **File Locations** dialog box to be displayed.

Step 6 At the **File Locations** dialog box, under **Destination...**, verify that the ORACLE_HOME directory path is correct. At **Name**, type a name (for

example, **ora9i)** and click **Next**. Clicking **Next** causes the **Available Products** dialog box to be displayed.

Step 7 At the **Available Products** dialog box, select **Oracle Cluster Manager 9.2.0.1.0** and click **Next**. Clicking **Next** causes the **Public Node Information** dialog box to be displayed.

Step 8 At the **Public Node Information** dialog box, type the management node name in one of the public node boxes, and the remaining cluster nodes. Click **Next**. Clicking **Next** causes the **Private Node Information** dialog box to be displayed.

Step 9 At the **Private Node Information** dialog box, type the management node name in one of the private node boxes, and the remaining cluster nodes.

Note If the node has two NICs, type the private alias in one of the private node boxes.

Click **Next**. Clicking **Next** causes the **WatchDog Parameter Information** dialog box to be displayed.

Step 10 At the **WatchDog Parameter Information** dialog box, use the default value and click **Next**. Clicking **Next** causes the **Quorum Disk Information** dialog box to be displayed.

Step 11 At **Quorum Disk Information** dialog box, type the full name of the raw device that has been mapped to the **oraraw1** quorum disk. Click **Next**. Clicking **Next** causes a **Summary** dialog box to be displayed.

Step 12 If the summary information is correct, click **Install**.

Step 13 Upon completion of installation, the **End of Installation** dialog box is displayed. Click **Exit** and proceed to the next section, "Configuring Cluster Manager."

Configuring Cluster Manager

To configure the Cluster Manager, follow these steps:

Step 1 Change the directory to **$ORACLE_HOME/oracm/admin**.

Step 2 Check to see that your system has a **cmcfg.ora** file. If there is no **cmcfg.ora** file, copy **cmcfg.ora.tmp** to **cmcfg.ora**.

Step 3 Check the **cmcfg.ora** file to make sure that the **HostName=** parameter is included in the file. If the **HostName=** parameter is not present, add the

parameter with its value equal to the private node name of the master node. For example, if the private node name of the master node was set to **racpvt1** (during installation of the Oracle Cluster Manager, "Installing Cluster Manager" on page 21), then the **HostName=** parameter should be set to **racpvt1**.

The following example shows **HostName=** set to **racpvt1**, the private node name of the master node:

```
### CMCFG.ORA ###
HeartBeat=15000
ClusterName=Oracle Cluster Manager, version 9i
PollInterval=1000
MissCount=20
PrivateNodeNames=racpvt1 racpvt2 racpvt3 racpvt4
PublicNodeNames=rac1 rac2 rac3 rac4
ServicePort=9998
#WatchdogSafetyMargin=5000
#WatchdogTimerMargin=60000
CmDiskFile=/dev/raw/raw1
→ HostName=racpvt1
```

Step 4 Edit the **ocmargs.ora** file to be like following sample file:

```
#Sample configuration file $ORACLE_HOME/oracm/admin/ocmargs.ora
oracm /a:0
norestart 1800
```

Note **oracm /a:0** is used for testing only. In a production environment, do *not* use the **/a:0** flag.

Step 5 Create your first set of Context Dependent Path Name (CDPN) directories.

a. As **oracle**, change directory to **$ORACLE_HOME**. Create a directory for each node in the cluster:

```
$ mkdir rac1 rac2 rac3 rac4
```

b. Copy the **oracm** directory into each of the node's directories, as follows:

```
$ for i in `ls -d rac*`; do; cp -r oracm $i; done;
```

c. Move the **oracm** directory to **oracm.orig** as follows:

```
$ mv oracm oracm.orig
```

d. To create the CDPN, enter the following:

```
$ ln -s @hostname/oracm oracm
```

e. Modify the **cmcfg.ora** file in the **oracm/admin** directory for each node. You only need to change the **HostName=** parameter to reflect the node you are logged in to.

f. At each node, log in to **root** and run the following:

```
# $ORACLE_HOME/oracm/bin/ocmstart.sh
```

g. Check for **oracm** processes

Step 6 Proceed to the next section, "Installing Oracle RDBMS."

Installing Oracle RDBMS

To install Oracle RDBMS, follow these steps:

Step 1 Start the Oracle Universal Installer (the same way as described in "Installing Cluster Manager" on page 21, Step 2). The **Cluster Node Selection** dialog box will be displayed.

Step 2 At the **Cluster Node Selection** dialog box, select the master node and click **Next**.

Note Because you are using a shared ORACLE_HOME, you only need to select the master node.

Step 3 Navigate to the **Available Products** dialog box, select **Oracle9i Database 9.2.0.1.0** and click **Next**. Clicking **Next** causes the **Installation Types** dialog box to be displayed.

Step 4 At the **Installation Types** dialog box, click **Custom** and **Next**.

Step 5 At the **Available Product Components** dialog box, ensure that **Oracle9i Real Application Cluster 9.2.0.1.0** is selected. Check other components for which you are licensed and click **Next**.

Step 6 At the **Shared Configuration File Name** dialog box, enter the second raw file pointing to the second quorum pool **oraraw2** and click **Next**.

Step 7 At the **Summary** dialog box, ensure that it shows that the Real Application Cluster software installs. Click **Install**.

Step 8 At the **Setup Privileges** dialog box (resulting from the previous step), you will be asked to log in to each node as **root** and run **$ORACLE_HOME/root.sh**. Click **OK** after you have verified that the file **/var/opt/oracle/srvConfig.loc** exists, and that it points to the quorum pool **oraraw2**.

Note You may get an error the first time you run **$ORACLE_HOME/root.sh**. If you encounter an error, rerun **$ORACLE_HOME/root.sh**.

Step 9 At the **End of Installation** dialog box, click **Exit**. You have completed the installation of software.

Step 10 Proceed to the next section, "Applying Patch."

Applying Patch

After installing the 9.2.0 version of Oracle, acquire the 9.2.0.3 (as of the writing of this document) patch for the Cluster Manager and RDBMS Server. Apply the patch as you did the major release (using the Oracle Universal Installer).

To apply the patch follow these steps:

Step 1 Shut down the Oracle Cluster Manager on each node.

Step 2 Remove the CDPN by performing the following, change to **$ORACLE_HOME**:

```
$ rm oracm
$ mv oracm.orig oracm
```

Step 3 Follow the instructions to apply the Oracle 9.2.0.3 patch.

Step 4 Apply the Cluster Manager software, and just enter the master node information in the private and public node entries. Finish the Cluster Manager installation.

Step 5 After the Cluster Manager software has been patched, move to the **$ORACLE_HOME/oracm/admin** directory and make the following modifications to the **cmcfg.ora** file:

- Raise the **MissCount** to 215.
- Remove the **WatchDog** parameters.
- Add a new variable, **KernelModuleName=hangcheck-timer**.

Note The reason for those changes is that the new Cluster Manager software uses the much improved GPL **hangcheck-timer**.

For example, see the following sample **cmcfg.ora** file:

```
### CMCFG.ORA ###
HeartBeat=15000
ClusterName=Oracle Cluster Manager, version 9i
PollInterval=1000
MissCount=215 <--------- Raised to 215
PrivateNodeNames=racnode1 racnode2 racnode3 racnode4
PublicNodeNames=rac1 rac2 rac3 rac4
ServicePort=9998
KernelModuleName=hangcheck-timer <-- Added, replacing Watchdog timer
CmDiskFile=/dev/raw/raw1
HostName=racnode1
```

Step 6 After modifying the **cmcfg.ora** file, recreate the CDPNs by running the following script from **$ORACLE_HOME**:

```
$ for i in `ls -d rac*`; do; cp -r oracm $i; done;
```

Step 7 Move the **oracm** directory to **oracm.orig** using the following command:

```
$ mv oracm oracm.orig
```

Step 8 To create the CDPN, enter the following:

```
$ ln -s @hostname/oracm oracm
```

Step 9 Restart the Oracle Cluster Manager on each node where Oracle Cluster Manager was stopped earlier.

Note Make sure that you modify the **cmcfg.ora HostName=** parameter to reflect the respective private node name before you start the software.

Step 10 Proceed to the next section, “Creating an Oracle Database.”

Creating an Oracle Database

Creating an Oracle database for use with Sistina GFS™ consists of the following tasks:

Step 1 Creating Network Configuration Files, page 29

Step 2 Enabling CDPN, page 32

Step 3 Setting Up the Cluster Configuration File, page 33

Step 4 Setting Up the Database, page 34

Creating Network Configuration Files

Creating the network configuration files consists of creating the following Oracle TNS-related files: **listener.ora**, **tnsnames.ora**, and **sqlnet.ora**.

Note The **$TNS_ADMIN** directory will eventually reside in CDPN, assuming that the default directory for **$TNS_ADMIN** is **$ORACLE_HOME/network/admin**.

To create the network configuration files, follow these steps:

Step 1 Create the **listener.ora** file; see Example 2 for a sample file.

Step 2 Create the **tnsnames.ora** file; see Example 3 for a sample file.

Step 3 Create the **sqlnet.ora** file; see Example 4 for a sample file.

Step 4 Proceed to the next section, “Enabling CDPN.”

Example 2 Sample File: listener.ora

```
listener_mydb1=
  (DESCRIPTION_LIST =
    (DESCRIPTION =
      (ADDRESS_LIST =
        (ADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC))
      )
      (ADDRESS_LIST =
        (ADDRESS = (PROTOCOL = TCP)(HOST = rac1)(PORT = 1521))
      )
    )
  )

SID_LIST_LISTENER_MYDB1 =
  (SID_LIST =
    (SID_DESC =
      (SID_NAME = PLSExtProc)
      (ORACLE_HOME = /mnt/oracle_base/product/9ir2)
      (PROGRAM = extproc)
    )
   (SID_DESC =
      (ORACLE_HOME = /mnt/oracle_base/product/9ir2)
      (SID_NAME = mydb1)
    )
  )
```

Example 3 Sample File: tnsnames.ora

```
MYDB.HQ.SISTINA.COM =
  (DESCRIPTION =
    (LOAD_BALANCE = ON)
    (FAILOVER = ON)
    (ADDRESS_LIST=

(ADDRESS=(PROTOCOL=TCP)(HOST=RAC1.HQ.SISTINA.COM)(PORT=1521))

(ADDRESS=(PROTOCOL=TCP)(HOST=RAC2.HQ.SISTINA.COM)(PORT=1521))

(ADDRESS=(PROTOCOL=TCP)(HOST=RAC3.HQ.SISTINA.COM)(PORT=1521))

(ADDRESS=(PROTOCOL=TCP)(HOST=RAC4.HQ.SISTINA.COM)(PORT=1521))
    )
    (CONNECT_DATA=
      (SERVICE_NAME=MYDB.HQ.SISTINA.COM)
      (failover_mode =
        (type = select)
   (method=basic)
   (retries=5)
   (delay=2)
```

```
      )
    )
  )

MYDB1.HQ.SISTINA.COM=
  (DESCRIPTION=

(ADDRESS=(PROTOCOL=TCP)(HOST=RAC1.HQ.SISTINA.COM)(PORT=1521))
    (CONNECT_DATA=
      (SERVICE_NAME=MYDB.HQ.SISTINA.COM)
      (INSTANCE_NAME=MYDB1)
      )
  )
MYDB2.HQ.SISTINA.COM=
  (DESCRIPTION=

(ADDRESS=(PROTOCOL=TCP)(HOST=RAC2.HQ.SISTINA.COM)(PORT=1521))
    (CONNECT_DATA=
      (SERVICE_NAME=MYDB.HQ.SISTINA.COM)
      (INSTANCE_NAME=MYDB2)
      )
  )
MYDB3.HQ.SISTINA.COM=
  (DESCRIPTION=

(ADDRESS=(PROTOCOL=TCP)(HOST=RAC3.HQ.SISTINA.COM)(PORT=1521))
    (CONNECT_DATA=
      (SERVICE_NAME=MYDB.HQ.SISTINA.COM)
      (INSTANCE_NAME=MYDB3)
      )
  )
MYDB4.HQ.SISTINA.COM=
  (DESCRIPTION=

(ADDRESS=(PROTOCOL=TCP)(HOST=RAC4.HQ.SISTINA.COM)(PORT=1521))
    (CONNECT_DATA=
      (SERVICE_NAME=MYDB.HQ.SISTINA.COM)
      (INSTANCE_NAME=MYDB4)
      )
  )
EXTPROC_CONNECTION_DATA.HQ.SISTINA.COM =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC))
    )
    (CONNECT_DATA =
      (SID = PLSExtProc)
      (PRESENTATION = RO)
    )
  )
```

Example 4 Sample File: sqlnet.ora

```
NAMES.DEFAULT_DOMAIN = hq.sistina.com
NAMES.DIRECTORY_PATH= (TNSNAMES)
```

Enabling CDPN

To enable CDPN, follow these steps:

Step 1 Change directory to **$ORACLE_HOME** and create your CDPN directories as follows:

```
$ for i in `ls –d rac*`; do; cp –r network $i; done;
```

Step 2 Move the network directory to **network.orig** using the following command:

```
$ mv network network.orig
```

Step 3 To create the CDPN, enter the following:

```
$ ln -s @hostname/network network
```

Step 4 *At each node*, navigate to the **$ORACLE_HOME/network/admin** directory and edit the **listener.ora** file to reflect the node-specific information.

Step 5 After you have edited the node-specific information in the **listener.ora** file, start your listener and test your configuration. If they all respond with **OK**, your network configuration is done; stop the listener.

Step 6 Make sure that the 9.2.0.3 patch properly sets the permissions for the **dbsnmp** file in the **$ORACLE_HOME/bin directory**. It should be owned by **oracle** and the permissions set at **750**.

Step 7 Proceed to the next section, “Setting Up the Cluster Configuration File.”

Setting Up the Cluster Configuration File

To set up the cluster configuration file, follow these steps:

Step 1 On each node, if the **srvconfig.loc** file does not exist in the **/var/opt/oracle** directory, log in as **root** and run the following script:

```
# cd $ORACLE_HOME/srvm/utl
# ./rootadd.sh
```

Note This step will add the **srvconfig.loc** file to the **/var/opt/oracle** directory. If it fails, run the commands in the script manually. Make sure to perform this step on each node.

Step 2 As **oracle** on the master node, configure the Oracle cluster information device according to the following example:

```
$ srvconfig -init
$ gsdctl start
$ srvctl add database -d mydb -o /mnt/oracle_base/product/9ir2
$ srvctl add instance -d mydb -i mydb1 -n rac1
$ srvctl add instance -d mydb -i mydb2 -n rac2
$ srvctl add instance -d mydb -i mydb3 -n rac3
$ srvctl add instance -d mydb -i mydb3 -n rac3
```

Step 3 Start the network listener on the master node as follows:

```
$ lsnrctl start listener_mydb1
```

Step 4 At each node, issue the following commands:

```
$ gsdctl start
$ agentctl start
```

Step 5 Proceed to the next section, “Setting Up the Database.”

Setting Up the Database

Before setting up the database, check the following:

- All the Oracle-required environment variables are set.
- **ORACLE_SID** is set to the instance name.
- Unset the **LANG** environment variable.

You can create a database manually or by using **dbca** that creates the scripts from which you can create the database. Create the database from the master node only.

To set up the database follow these steps:

Step 1 Modify the **init.ora** parameter to comment out the **local _listener** that is not appropriate to the node from which you you are working. See Example 5 for a sample of an **init.ora** file.

Step 2 Set the environment variables. For example:

```
$ export ORACLE_BASE=</mnt/oracle>
$ export  ORACLE_HOME =$ORACLE_BASE/product/9iR2
$ unset LANG
$ export ORACLE_SID=mydb1
```

Step 3 Assuming that your database conforms to the Oracle Optimal Flexible Architecture (OFA) guidelines, and that **bdump**, **udump**, **cdump**, **pfile**, and any other relevant directories are under **$ORACLE_BASE/admin**, put **$ORACLE_BASE/admin** in CDPN:

```
$ cd $ORACLE_BASE
$ for i in `ls -d product/9ir2/rac*`; do; cp -r admin $i; done;
$ mv admin admin.orig
$ ln -s product/9ir2/@hostname/admin admin
```

Step 4 Create the password files, **spfiles**, from the respective nodes/directories.

Step 5 Start the database instances from the respective nodes using **srvctl** (the preferred method) or SQL*Plus.

a. Using **srvctl—**For example:

```
$ srvctl start instance -d mydb -i mydb2
```

b. Using SQL*Plus—For example:

```
$ sqlplus /nolog
SQL> connect / as sysdba
SQL> startup
ORACLE instance started.

Total System Global Area  252777144 bytes
Fixed Size                   451256 bytes
Variable Size             218103808 bytes
Database Buffers           33554432 bytes
Redo Buffers                 667648 bytes
Database mounted.
Database opened.
```

Step 6 Start the listeners from the respective nodes.

Example 5 Sample File: init.ora

```
...
...
cluster_database_instances = 4

...
cluster_database=true
mydb1.instance_name=mydb1
mydb1.instance_number=1
mydb1.local_listener=LISTENER_MYDB1
mydb1.local_listener=mydb1.hq.sistina.com
mydb1.thread=1
mydb1.undo_tablespace=UNDOTBS1
mydb2.instance_name=mydb2
mydb2.instance_number=2
→ #mydb2.local_listener=LISTENER_MYDB2
mydb2.local_listener=mydb2.hq.sistina.com
mydb2.thread=2
mydb2.undo_tablespace=UNDOTBS2
mydb3.instance_name=mydb3
mydb3.instance_number=3
→ #mydb3.local_listener=LISTENER_MYDB3
mydb3.local_listener=mydb3.hq.sistina.com
mydb3.thread=3
mydb3.undo_tablespace=UNDOTBS3
mydb4.instance_name=mydb4
mydb4.instance_number=4
→ #mydb4.local_listener=LISTENER_MYDB4
mydb4.local_listener=mydb4.hq.sistina.com
mydb4.thread=4
mydb4.undo_tablespace=UNDOTBS4

...
...
```

Index