CHAPTER 12

Oracle Real Application Clusters

This chapter in the high availability section is devoted to the discussion of Oracle Real Application Clusters. Real Application Clusters (RACs) constitute the next generation of Oracle Parallel Server—it is technology that has expanded beyond the original capability of Parallel Server to allow easier shipment of blocks between instances, reducing the additional I/O normally associated with Parallel Server. At the same time, RAC maintains the *cache coherency* needed for multiple nodes to be able to read the most current data. It is beyond the scope of this book to go into a full description of Real Application Cluster architecture—the Oracle9*i* documentation set has a total of six different books devoted to various RAC-related topics. This chapter essentially presents a cookbook approach to RAC to allow you to get up and running as quickly as possible in a RAC environment on Windows 2000. We will disseminate the following tips:

- Understanding the RAC environment
- Using the interconnect
- Discovering scalability advantages in a RAC environment
- Configuring network cards
- Configuring the interconnect
- Creating and managing raw partitions for your files
- Using the clustercheck tool
- Manually installing Object Link Manager prior to cluster setup
- Creating symbolic links
- Exporting and importing link definitions
- Initiating the Cluster Setup Wizard
- Defining symbolic links within the cluster setup
- Installing Oracle RAC software
- Creating a RAC database
- Maintaining a RAC database
- Using system managed undo in a RAC database
- Using multiple redo threads
- Adding additional instances
- Adding additional datafiles and log groups

The RAC Environment

The hardware setup used for Real Application Clusters on Windows 2000 is similar to what you would use for Oracle Failsafe—it consists of two or more nodes, is connected to the same shared disk array, and includes an interconnect for private communications between the nodes. Like Failsafe, this interconnect normally consists of a private, dedicated network between the nodes. However, there are differences in the way the shared drives are accessed, how the interconnect is used, and in the clustering software.

Shared Everything in the RAC Environment

Because of the shared-everything concept, a RAC setup differs from an Oracle Failsafe setup in many ways. To begin with, a RAC environment puts a much greater load on the interconnect. In addition to checking the status of each node, the interconnect is used for shipping data blocks between nodes for cache coherency. Oracle's term for this mechanism is *cache fusion*. This essentially means that, whenever possible, data blocks move between each instance's cache without needing to be written to disk, with the key being to avoid additional I/O being necessary to synchronize the buffer caches of multiple instances.

Importance of the Interconnect

This traffic is directed to go through the interconnect, with the expectation that this will be significantly faster than going to disk. To meet this expectation, you must ensure that the hardware you have dedicated to your private network is capable of meeting the demands for throughput that will be placed upon it. A gigabit Ethernet connection is recommended. Also, since the interconnect provides an even more crucial role, you are more likely to require redundancy in this area. In addition, new methodologies for maintaining cache coherency are evolving. One such methodology is VIA, or Virtual Interface Architecture (discussed in Chapter 7), which is essentially a technology standard that has evolved specifically for clustered environments. This standard calls for a simple hardware implementation for reading data structures in a user's memory space and moving them directly to the user memory space on another node. Oracle9*i* Real Application Clusters provides support for VIA, but you must contact your hardware vendor for configuration and certification information. This chapter will focus on the more common Ethernet connection for the private network.

Availability and Scalability in a RAC Environment

Because there are multiple nodes, RAC still gives you crash recovery and Transparent Application Failover (TAF) capabilities, similar to what you have in an Oracle Failsafe environment: when one node goes down, the other node continues processing. (We discuss TAF in more detail in Chapter 15.) As discussed in the previous chapter, disaster

recovery capability is still a function of the hardware and how much separation you can attain between nodes and disks and disk mirrors. In addition to crash recovery or disaster recovery capability, a RAC environment also affords scalability. This is because all nodes are simultaneously accessing the shared disk, which contains the database. With this shared-everything architecture, the horsepower of multiple nodes, in terms of memory, processing power, and networking capacity, can all be put into play all at once, vastly increasing throughput and the number of concurrent users. This is one solution to the per-process memory limitations mentioned in Chapter 5.

Use Raw Partitions

Aside from differences in how the private network is used, the shared-everything architecture requires you to view the shared drives in a totally different light. Because all nodes are accessing the disk concurrently, you cannot rely on a normal file system such as NTFS to maintain access to these drives and avoid disk or data corruption. Instead, this is accomplished by leaving the drives unformatted, or raw. This allows the *distributed lock manager (DLM)* within the Oracle RDBMS to control access to the data blocks, ensuring that only one node is writing a given block at any given time.

Since there is no file system on a raw partition, there can only be *one* file per partition, and all database files must be on a raw device—including datafiles, control files, redo logs, and even the SPFILE. An exception to this is the archive logs, which must be written to a file system. This need for all files to be on their own raw partition results in the necessity to spend much forethought and time in laying out and partitioning the disks for all of the various files required by the database. Once the disks are partitioned, Oracle accesses the raw partition by virtue of a symbolic link, mapping a link name to a physical disk number and partition number. This work must be done prior to the installation of the cluster software, which in turn must be completed before RAC can be installed.

RAC Cluster Software

Which brings us to the next topic—the cluster software. The cluster software for a RAC environment must be installed and running prior to installing Oracle Enterprise Edition. Otherwise, the Real Application Clusters option will not be available for installation. Here, we will discuss tips regarding the RAC software requirements to be aware of when planning your setup. Later in this chapter, we will discuss the actual installation of the cluster software itself.

No Virtual Groups

As we noted in the previous chapter, Real Application Clusters do not use the MSCS software, except in the case of Real Application Clusters Guard, discussed in Chapter 15. As such, unless you are using Real Application Clusters Guard,

there is no concept of a virtual server or virtual groups. Since both nodes are accessing the disks, and hence the database, simultaneously, a virtual group is not needed—connections can be made to either node. Also in Chapter 15, we will discuss Transparent Application Failover, or TAF, which is a method of configuring SQLNET files so that a client can connect to any node in the cluster or cause failover to any node in the cluster transparently (that is, with no end-user intervention required).

Vendor-Provided Versus Oracle-Provided OSD Software

In releases prior to Oracle9*i* on the Windows NT/2000 platform, sites that used Oracle Parallel Server relied on the hardware vendors of certified platforms to provide and support the cluster, or OSD, software. This is no longer the case with Oracle9*i*—Oracle now provides the OSD software on the 9i CD-ROM, and Oracle also provides the necessary support. However, the certification requirements are not changed. Even though you have ready access to the clustering software, you must still ensure that you are running RAC on a hardware platform that your vendor has certified for the version of RAC and the version of the OS that you are running on. Oracle may still support third-party OSD software, but it will have to be certified to run Real Application Clusters, and in the case of third-party OSD software, Oracle will not support the cluster configuration. To obtain the latest list of certified hardware and software combinations, you should contact Oracle Support or your hardware vendor.

Voting Disk

Similar to the quorum disk used by MSCS, the RAC cluster software requires that one of the shared drives be configured as a voting disk. This is needed to resolve any conflicts between nodes, as an alternate communication means that all nodes can access should the interconnect fail. In addition, this disk contains information on the instances and nodes for each database in your cluster, as well as the ORACLE_HOME for the database. This information is read by the DBCA, Enterprise Manager, and other Oracle tools.

Preparing for the Cluster Installation

Configuring the operating system environment consists of a couple of different steps that must be taken before you even begin to install the clustering software or the RAC option. First, you must ensure that the interconnect is configured correctly. Second, you must partition disks according to how they will be laid out for your database, keeping in mind that each file requires its own partition. Finally, you must define the links for these raw partitions. This last step of configuring the symbolic links can be done prior to the cluster installation using the Oracle Object Link Manager, or it can be done during the cluster installation using the Cluster Setup Wizard. We will discuss both methods in detail in this chapter.

Configuring the Interconnect

Configuring the interconnect in a RAC environment is almost identical to what we described in Chapter 11 for Oracle Failsafe. You will most likely have at least two network cards on each node. Again, you will want to ensure that the card you assign the *public* IP address to is the card that is bound first, and the card with the *private* IP address is bound last. (Refer to the sections "Configuring Network Cards" and "Binding Order" in Chapter 11 for more details on this setup.) Since there is the potential for high traffic going across the interconnect in a RAC environment, you want to ensure that you provide the fastest possible connection, as noted previously. Also, since the private network card is usually not connected to a DNS server, you should define a network name for the private IP in the HOSTS file of each node (the HOSTS file is found in \WINNT\system32\drivers\etc, as discussed in Chapter 11). Again, use the convention of <nodename>.SAN for the private host name:

```
127.0.0.1          localhost
10.10.10.1          RMNTOPS1.SAN
10.10.10.2          RMNTOPS2.SAN
```

Configuring the Raw Partitions

As in the case of Oracle Failsafe, you will also need to configure the shared drives and partition the drives according to your needs. However, there is a huge difference in how many partitions you configure, since you must have one shared partition for each file. First, you must determine how many shared physical devices you have available. Next, determine how many of them can be used for the RAC database that you intend to create. In Figure 12-1, you see that we have Disks 0 through 8, for a total of nine physical drives. Of these, three have been labeled as private, and three have been labeled as shared, but they are already formatted NTFS (this is not required—we have it configured thus because we intend to install Microsoft Cluster Server and Real Application Cluster Guard later). That leaves us with three additional shared drives on which we can place datafiles for our RAC database.

Link Names and Partitions Required by the Database Assistant

Recall from the previous section that all datafiles, control files, online redo logs, and even the SPFILE must be on the shared drive, and each one must have its own partition. If you use the DBCA (which will be kicked off automatically after the installation), the Database Assistant will expect a certain number of partitions to have been created already. In addition, it will also anticipate that certain link names have been defined

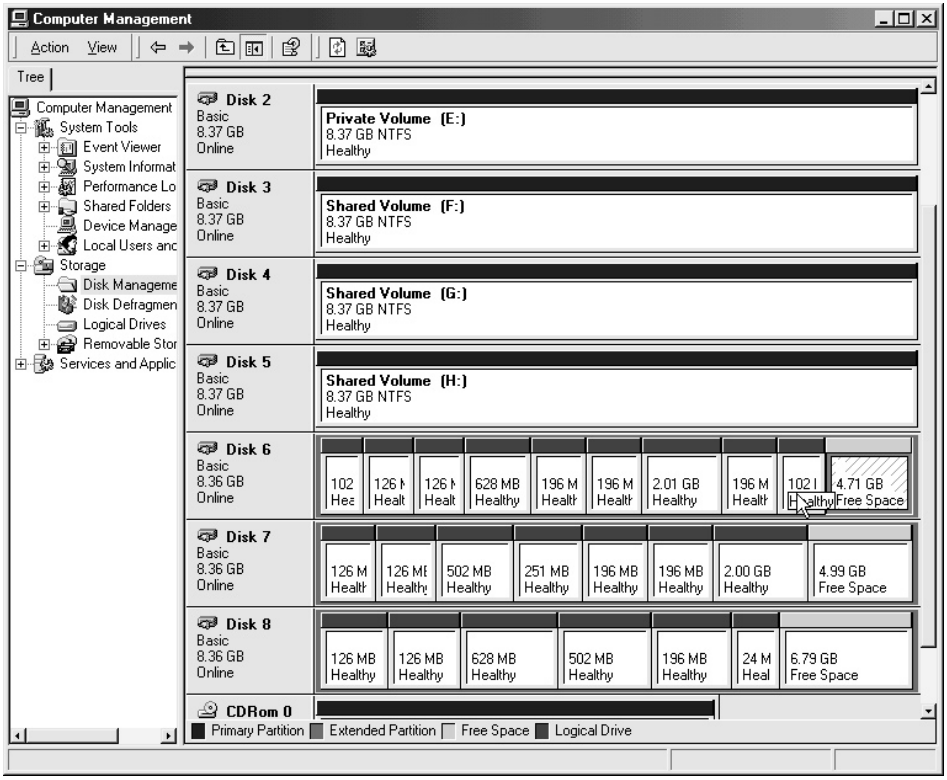


FIGURE 12-1. *Disk Manager view of available drives*

for each partition. The link name should always be prefaced with the global database name of the database you are creating. We recommend that you lay out a table similar to Table 12-1 to help you determine how many partitions to create, and on which devices they should go. In Table 12-1, you see that the DBCA anticipates at least 17 partitions. One partition is for the SPFILE, and two partitions are for control files. Each instance must have its own online redo log group, so there are a total of four online logs (we have only two nodes). In addition, if you are using system managed undo, each instance requires its own undo tablespace. Finally, you have these tablespaces: system, temporary, drsys, cwmlite, example, users, index, and tools. An eighteenth partition is required as a voting disk, or quorum disk, to allow the cluster to resolve ownership of the disks in case the interconnect should fail.

So, assuming that you are creating a database with the name RACDB, you would need partitions of the names and sizes shown in Table 12-1.

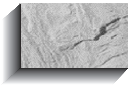
Link Name	File Type	Min Size	Disk Number	Partition Number
SRVCFG	Voting disk	100MB		
RACDB_SPFILE1	SPFILE	25MB		
RACDB_CONTROL1	Control file 1	125MB		
RACDB_CONTROL2	Control file 2	125MB		
RACDB_REDO1_1	Instance 1 redo 1	125MB		
RACDB_REDO1_2	Instance 1 redo 2	125MB		
RACDB_REDO2_2	Instance 2 redo 1	125MB		
RACDB_REDO2_1	Instance 2 redo 2	125MB		
RACDB_SYSTEM1	System tablespace	500MB		
RACDB_UNDOTBS1	Instance 1 undo tablespace	625MB		
RACDB_UNDOTBS2	Instance 2 undo tablespace	625MB		
RACDB_TEMP1	Temporary tablespace	500MB		
RACDB_USERS1	Users tablespace	100MB		
RACDB_INDX1	Index tablespace	50MB		
RACDB_DRSYS1	Intermedia tablespace	250MB		
RACDB_TOOLS1	Tools tablespace	50MB		
RACDB_CWMLITE1	OLAP tablespace	200MB		
RACDB_EXAMPLE1	Example schemas tablespace	150MB		

TABLE 12-1. *Determining Numbers and Sizes of Partitions to Create*

These values in Table 12-1 are minimum sizes for using the Data Warehousing template. Other templates used by the DBCA will work with the same links, but you can get by with a smaller temporary tablespace and a smaller undo tablespace. On the other hand, you may decide that you want to create larger partitions for your temporary or undo tablespaces, and/or larger partitions for the online redo logs. It makes life easier to plan for these things up front. To expand these tablespaces later, or to add additional online redo logs, you will have to create a new partition, and a new link, so that you can then add a second datafile to a tablespace. In addition to the partitions we've just outlined, you will want to include partitions for any additional tablespaces that you require for your own application, and any additional redo log groups that you need.

Mapping Link Names to Devices

Once you have determined the number and sizes of partitions that you will need, you must determine next how you are going to spread them across the available shared disks. Using Table 12-1, fill in disk numbers first to give yourself an idea of what types and numbers of files will be on each disk. Next, go back and assign the partition numbers of the partitions that you will create, starting with number 1 on each disk.



CAUTION
Never assign a symbolic link name to partition 0. Partition 0 is used by Windows 2000 to write the signature on the disk. Always start counting at partition number 1.

The result of this exercise will end up looking something like what we have in Table 12-2 (note that this table has been sorted by the disk number).

Link Name	File Type	Min Size	Disk Number	Partition Number
SRVCFG	Voting disk	100MB	6	1
RACDB_REDO1_1	Instance 1 redo 1	125MB	6	2
RACDB_REDO2_1	Instance 2 redo 2	125MB	6	3
RACDB_USERS1	Users tablespace	100MB	6	4
RACDB_INDX1	Index tablespace	50MB	6	5
RACDB_UNDOTBS2	Instance 2 undo tablespace	625MB	6	6

TABLE 12-2. *Determining Locations of Partitions*

Link Name	File Type	Min Size	Disk Number	Partition Number
RACDB_REDO1_2	Instance 1 redo 2	125MB	7	1
RACDB_SYSTEM1	System tablespace	500MB	7	2
RACDB_DRSYS1	Intermedia tablespace	250MB	7	3
RACDB_TOOLS1	Tools tablespace	50MB	7	4
RACDB_CONTROL1	Control file 1	125MB	7	5
RACDB_CWMLITE1	OLAP tablespace	200MB	7	6
RACDB_CONTROL2	Control file 2	125MB	8	1
RACDB_REDO2_2	Instance 2 redo 1	125MB	8	2
RACDB_UNDOTBS1	Instance 1 undo tablespace	625MB	8	3
RACDB_TEMP1	Temporary tablespace	500MB	8	4
RACDB_EXAMPLE1	Example schemas tablespace	150MB	8	5
RACDB_SPFILE1	SPFILE	25MB	8	6

TABLE 12-2. *Determining Locations of Partitions* (continued)

Creating the Actual Partitions

You are now ready to create the actual partitions. Go to Disk Management in the Computer Management Console and highlight the first shared drive to be partitioned. In our case, this is disk number 6. Be sure that each disk that you intend to use is defined as a basic disk. Dynamic disks are not supported in a RAC environment. For each shared disk, create an extended partition that is equal to the entire size of the disk (do not create primary partitions, as you are limited to how many primary partitions you can have on a machine). Do this by right-clicking the disk itself, and choosing Create Partition. When prompted for the partition type, choose Extended Partition and then use all of the available space. Do this for each disk listed in the Disk Number column.

Creating Logical Drives

Once you have created an extended partition on the disk, go back and right-click the disk again. This time, choose Create Logical Drive (see Figure 12-2). Enter the size that you want for the first partition (corresponding to the Partition Number column in Table 12-2). On the next screen, choose the option Do not assign a drive letter or

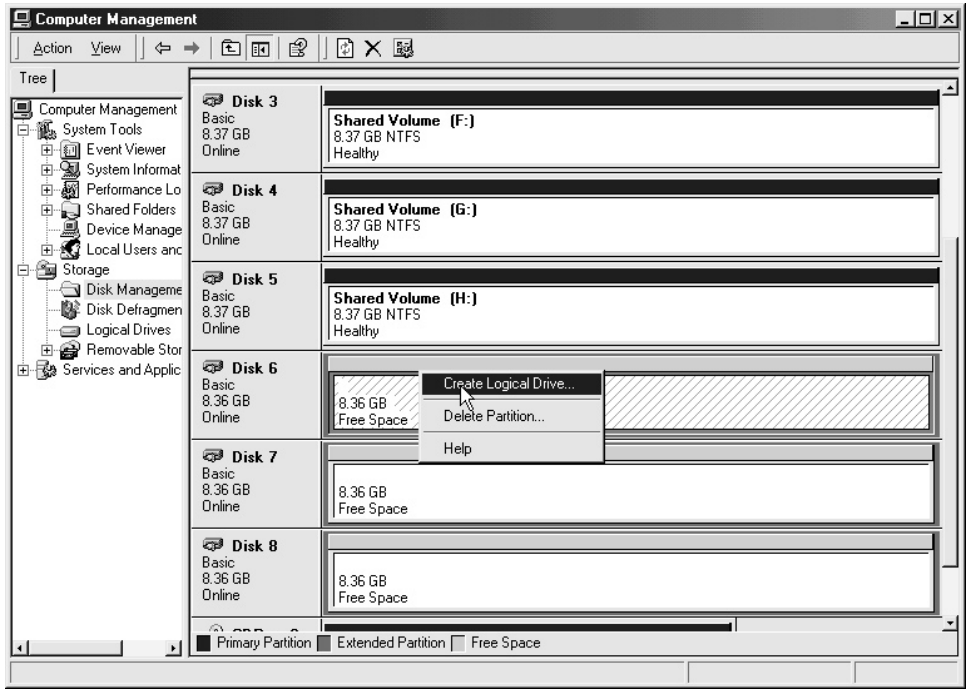


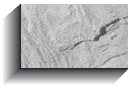
FIGURE 12-2. *Creating a logical drive*

path. Next, choose the option Do not format this partition, and finally click Finish on the last screen. Repeat these steps for each of the partitions listed in Table 12-2.

Removing Drive Letters The steps in the preceding section should be performed from one node only. You will find that once these partitions are created, even though you elected not to assign a drive letter, drive letters will have been assigned on the additional nodes. You can remove these drive letters on the other nodes through Disk Manager by right-clicking each drive, choosing Change Drive Letter and Path, and then choosing Remove. This is very cumbersome, however. To ease this process, Oracle provides a utility called LetterDelete, which can be run on each node to remove drive letters on all raw partitions. This utility will be found on Disk 1 of the Oracle9*i* CD-ROM set, located in the subdirectory Disk1\preinstall_rac\olm. Simply run the utility from a command prompt as follows:

```
C:\preinstall_rac\olm>letterdelete
Letter Delete: About to delete all drive letters on Oracle Partitions.
 Are you sure (y/n)?y
```

```
Deleted J:\ on \Device\Harddisk6\Partition1
Deleted K:\ on \Device\Harddisk6\Partition2
Deleted L:\ on \Device\Harddisk6\Partition3
Deleted M:\ on \Device\Harddisk6\Partition4
Deleted N:\ on \Device\Harddisk6\Partition5
Deleted O:\ on \Device\Harddisk6\Partition6
Deleted P:\ on \Device\Harddisk6\Partition7
Deleted Q:\ on \Device\Harddisk7\Partition1
Deleted R:\ on \Device\Harddisk7\Partition2
Deleted S:\ on \Device\Harddisk7\Partition3
Deleted T:\ on \Device\Harddisk7\Partition4
Deleted U:\ on \Device\Harddisk7\Partition5
```



NOTE

As evidenced by the preceding warning, all drive letters will be removed from any unformatted (raw) partitions. If you determine that you need a drive letter on one of these partitions, you can reassign the drive letter afterward to the partition individually after running LetterDelete.

Creating Symbolic Links

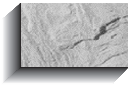
Symbolic links are the method by which Oracle accesses the raw partition. Without a drive letter and a file system allowing you to track directory names and filenames, you must somehow be able to tell the database where to find its files. This is done by creating a symbolic link with a name that, in translation, says something like, "Go to hard drive number 6, partition number 3, to find my data." Symbolic links are referenced within Oracle as \\.\<link_name>. In the next section, we will show you how to create these link names.

Manually Installing the Object Link Manager

For the example in the previous section, we have copied the contents of the preinstall_rac\olm directory to the local hard drive. Aside from the LetterDelete utility, there are several other useful utilities in this directory. One of these utilities is the GUI Object Link Manager, run by the executable GUIOracleOBJManager.exe. Once your drives are partitioned, this is the utility you use to create the symbolic links, telling Oracle how to access the files on these partitions. Before you can use this utility, you must install the Oracle Object Service by running the following command on each node of the cluster:

```
C:\ preinstall_rac\olm> oracleobjservice /install
```

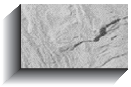
After installing the service, ensure that it is started on each node by checking the Services Console, and start it on each node if necessary. Once you have started the service, you will be able to proceed to the next step, creating symbolic links from one node and syncing them automatically on all nodes.



NOTE
This command is run prior to the installation of the cluster software. With this method, we are defining the links prior to installing the cluster software. If you do not run this command now, the Object Link Manager Service is installed automatically when the cluster server is installed. This will require that you define the links during the cluster setup. We will discuss how to do so later in this chapter. The Oracle Object Service can be removed by running the command ***oracleobjservice /remove****.*

Creating the Links

Once the service is running, you can now run the Link Manager. Do this by double-clicking GUIOracleOBJManager.exe in the preinstall_rac\olm directory. You will see a screen similar to the one shown in Figure 12-3, which displays the hard disk number, partition number, and partition size of the shared partitions it has found on the cluster. To assign a link name, right-click under the column called New Link Name. Referring back to Table 12-2, enter the link name assigned to that particular hard drive and partition. Remember that you must have a link called srvcfg for the voting disk, and the rest of the links must be prefaced with the global name of your database. Once you have entered all of the link names under the New Link Name column, choose Commit from the Options menu, and then choose Sync Nodes. You should now be able to see all links under the Oracle Link Name column, on all nodes.



CAUTION
Placing a check mark in the box next to a link name will result in the link name being deleted once you perform a commit operation.

Exporting and Importing Links

Once the links have been created, you should export them to a file as a backup. Again, there is a utility in \preinstall_rac\olm to allow you to do this—the ExportSymLinks utility. Export symbolic links to a file with a .tbl extension using the /F: switch, as in

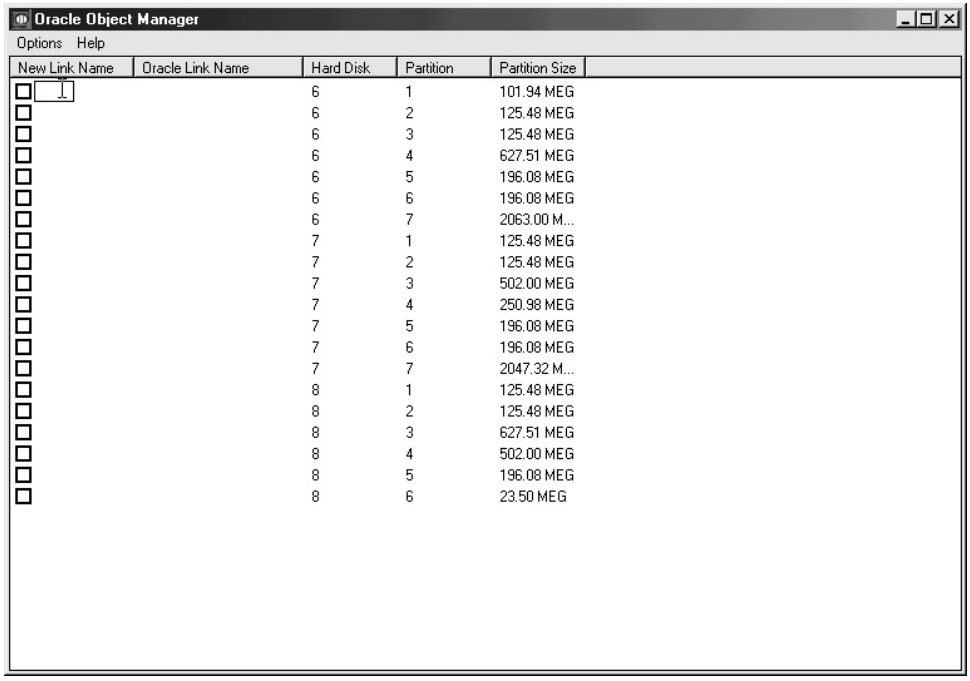


FIGURE 12-3. *Creating links in Object Link Manager*

the following example (note that /F: does *not* refer to a drive letter—this is a required switch indicating a filename follows):

```
C:\preinstall_rac\olm>exportsymlinks /F:D:\backups\racdblinks.tbl
Symbolic Link Exporter
Version 2.0.1
Copyright 1989-2001 Oracle Corporation. All rights reserved.

 Links exported to file D:\backups\racdblinks.tbl
 ExportSYMLinks completed successfully
```

If you should ever need to import the links back to reassign names to partitions, rather than retyping everything in the Object Link Manager window, you can now run the ImportSymlinks utility:

```
C:\preinstall_rac\olm>importsymlinks /F:D:\backups\racdblinks.tbl
```

Additional Utilities for Managing Raw Partitions

As you can see, there is a plethora of handy little utilities in the preinstall_rac\olm directory. A couple of other utilities that are now available are crlogdr and logpartformat. Once an extended partition has been created, crlogdr can be used from the command line to create logical drives on a given physical drive number (as opposed to using the Disk Management console). For usage, simply run crlogdr.exe by itself from a command prompt. The logpartformat utility is used to "format" a raw partition. Obviously, it does not truly format the partition with a file system, but it does clean up any stray bytes on the partition. If you are testing and end up deleting and re-creating files on these partitions on a regular basis, it is recommended that you run the logpartformat utility to clear the partition of any junk that may have been on there previously, prior to placing files for your production system. To run logpartformat, simply pass the link name of the partition you wish to format:

```
C:\preinstall_rac\olm>logpartformat racdb_junk
Logical Partition Formatter
Version 2.0
Copyright 1989-2001 Oracle Corporation. All rights reserved.

The logical drive racdb_junk will be formatted.

Formatting the logical drive WILL result in loss of ALL data.
Are you sure you want to continue?...(Y/N) y
```


Running the RAC Clustercheck

One final check should be run on your cluster prior to installing the cluster software. This is done with the clustercheck utility, found on the CD-ROM in the \DISK1\preinstall_rac\Clustercheck directory. This utility is similar in concept to the Verify Cluster tool that we use in a Failsafe environment, in that it is checking the health of the cluster components involved, including the interconnect and the available shared drives. However, it differs in that it is run *prior* to installation, and is therefore not meant to check the health of Oracle installation itself. Kick it off by running clustercheck.exe:

```
C:\preinstall_rac\clustercheck>clustercheck
```

Defining Public and Private Node Names for Clustercheck

You will be prompted for the number of nodes that are in the cluster and the public names of those nodes, as you see in Figure 12-4. Notice that the utility asks for the host name for node 1 and then for node 2. The host name that you specify for node 1 should be the host from which you eventually run the Cluster Setup Wizard and the Oracle installation. Clustercheck will then verify that the host name given can be successfully resolved to an IP address, and you will be asked to confirm the results.

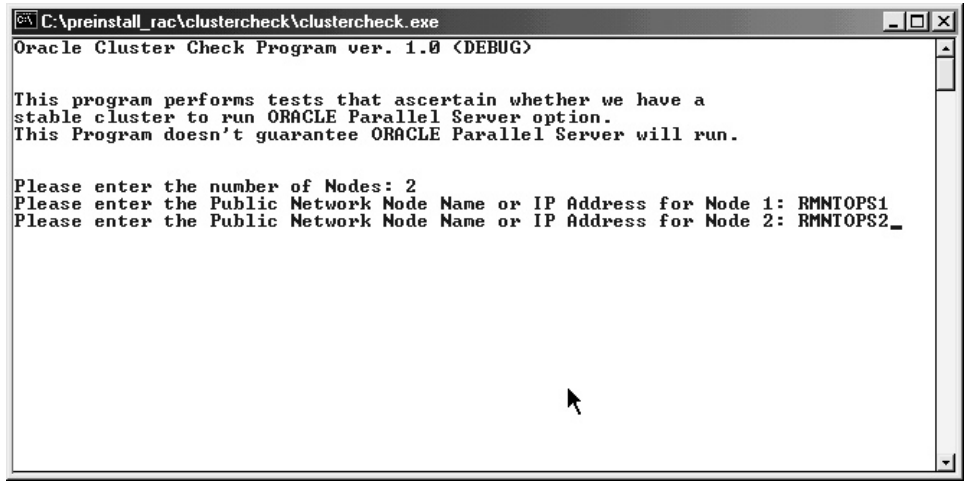


FIGURE 12-4. *Public node names in clustercheck*

Next, you are prompted for information on the private interconnect. Put in the private network names, as defined in the host's file in the earlier section "Configuring the Interconnect." In our example, these names are RMNTOPS1.SAN and RMNTOPS2.SAN (see Figure 12-5). You will then be asked to confirm the IP information for the private interconnect as well.

Clustercheck Log Files

During the clustercheck run, a service called InfoGatherer will be created on each node—this service will write log files into your temp directory, and then it will be deleted as quickly as it was created. The log files left behind by the InfoGatherer Service hold the key to determining if the clustercheck operation was successful. To find these logs, you must determine what Temp is set to. Right-click My Computer on the desktop, and choose Properties | Advanced | Environment Variables. Look under User Variables. By default, you will see something like C:\Documents and Settings\Administrator\Local Settings\Temp, where Administrator is the name of the user account you are logged in under. In this directory will be a subdirectory called OPSM, which is where the log files from the InfoGatherer Service are located.

The clustercheck utility will make sure that it has permissions to write to drives on all nodes and open Registry keys on all nodes. It will also verify access and check the health of the shared drives. Check these logs carefully for any indication of a problem. If you see a message similar to this at the bottom of OraInfoCoord.log,

```
ORACLE CLUSTER CHECK WAS SUCCESSFUL
```

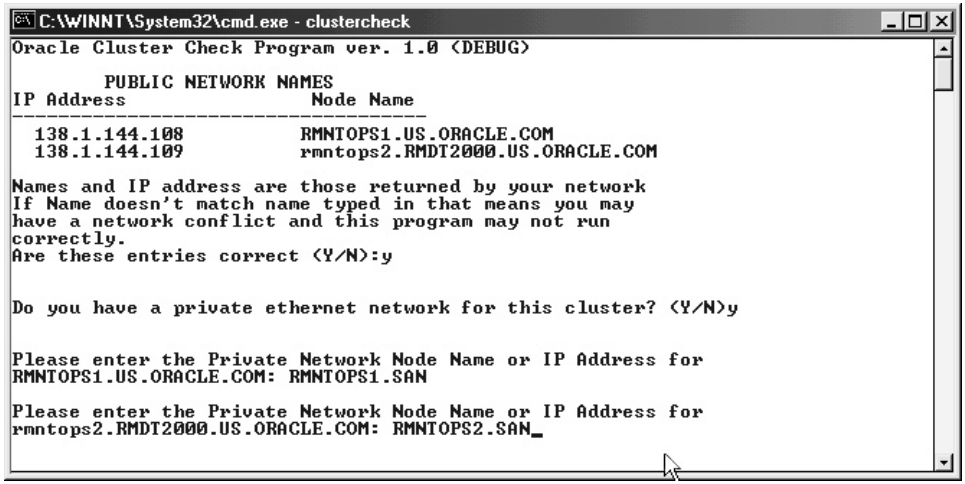


FIGURE 12-5. *Defining the private network in clustercheck*

then chances are good that you can proceed with the cluster setup. Otherwise, troubleshoot any permissions problems or problems with the shared drives, and then rerun the clustercheck utility.

Installing the Cluster Software and Oracle9*i* RDBMS

Once you have completed all of the necessary preinstallation work for the cluster, now comes the moment you have been waiting for—the installation of the cluster software for Real Application Clusters. Drumroll please! Okay, forget the pomp and circumstance. This next section will cover the final preinstallation piece necessary prior to installing the actual Oracle RDBMS. As noted previously, Oracle may support clustering software from other third-party vendors in order to run RAC, but it will not support the installation and configuration of another vendor's product. Thus, this section is devoted to the installation and configuration of the Oracle-provided cluster software only.

Running the Cluster Setup Wizard

Like just about everything else we have discussed so far in this chapter, the Cluster Setup Wizard is found in the \preinstall_rac directory, in a subdirectory named, appropriately enough, clustersetup. Previously, we showed you how to run many of

the utilities by simply copying the contents of the preinstall_rac directory to the local machine, and kicking off everything from there. You will find, however, that the Cluster Setup Wizard must be run from the actual CD-ROM or staging area because it makes use of the Oracle Universal Installer. Therefore, unless you run it from the CD-ROM or staging area, it will not be able to kick off. If you run Cluster Setup Wizard and briefly see a command prompt window and then nothing, verify that you are running it from the CD-ROM or the staging area.

Creating the Cluster

When you first run the Cluster Setup Wizard, the second screen of the Oracle Cluster Setup Wizard will only give you the option to create a cluster. On subsequent runs of the wizard, you will have the option to add additional nodes. It will be necessary to run this wizard again if you expand in the future. With the Create A Cluster option selected, click Next and continue on. If you have not yet partitioned any drives, you will not be able to continue past this point; instead, you will see the error in Figure 12-6. This is because you must have at a minimum one shared partition to be used for the voting disk. If you receive this error, you must exit and revisit the section "Creating Logical Drives," earlier in this chapter.

Assuming you make it to the next screen, you must now select the partition you want to represent the voting disk. Highlight the correct partition. In our example, shown in Figure 12-7, all of the link names and partitions are already filled in because we installed the Oracle Object Link Manager and defined them as described in the previous section. If you have not done this, the next section will walk you through defining the symbolic links using the Cluster Setup Wizard.

Defining Links in Cluster Setup Back up for just a moment and assume that we did not have the links predefined. If this were the case, the Symbolic Link column will be blank in the Cluster Setup Wizard. In such a scenario, you would click the Create Oracle Symbolic Links button. This actually calls the Oracle Object Link Manager,

FIGURE 12-6. *Cluster Manager error generated if no raw partitions exist*

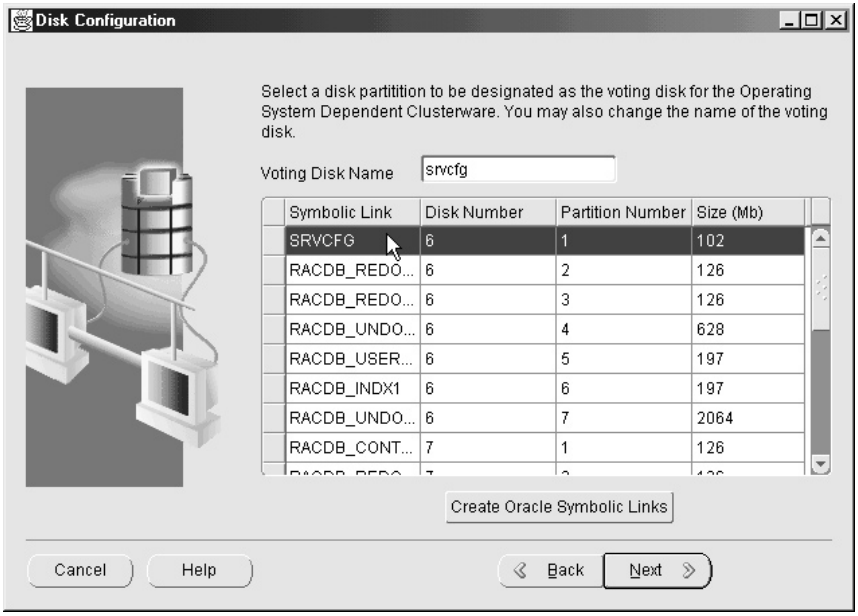


FIGURE 12-7. *Cluster Manager with symbolic links predefined*

with a slightly different interface. A screen like Figure 12-8 will appear. Simply enter the link names from Table 12-2, based on the disk and partition number associated. Once you have typed in all of the names, click Apply, and then close the window.

Defining the Network for the Interconnect

On the next screen, Oracle will check for the existence of VIA. If it is detected, you will be prompted to use that for the interconnect. If not, the setup will continue, and you will be asked which network you want to use for the interconnect. Choose the Private option and continue on. You will now be prompted for the public and private names assigned to each node. Fill them in as shown in Figure 12-9, using the same names defined in the local hosts file for the private name as were used when running the clustercheck utility. If you are using the convention of <nodename>.san, the private names will be filled in for you automatically. Once you have filled in the names, click Next. Note that the default location to install the files is not in the ORACLE_BASE directory structure; instead, the files will go in \WINNT\System32\osd9i. We recommend that you accept this default location and finish the installation.

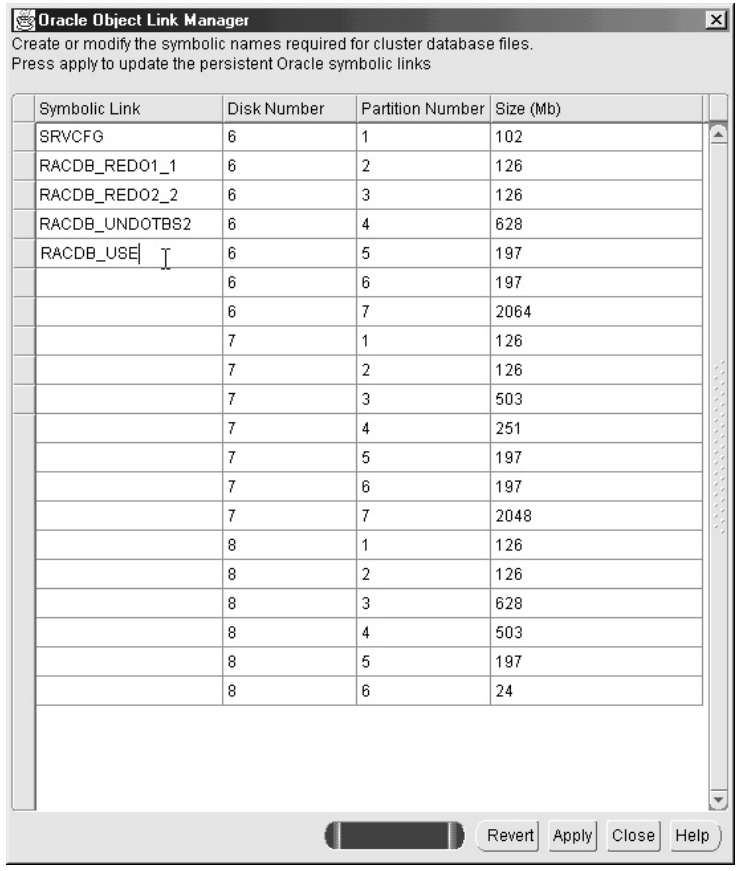


FIGURE 12-8. *Oracle Object Link Manager as seen during cluster setup*

What Just Happened?

When the installation is complete, you may want to investigate the changes made to your system. Recall that we recommended you run the Cluster Setup Wizard from what you defined as node 1 when you ran clustercheck. You only need to run the setup from this node. It will use the interconnect to write to the Registry and copy files over

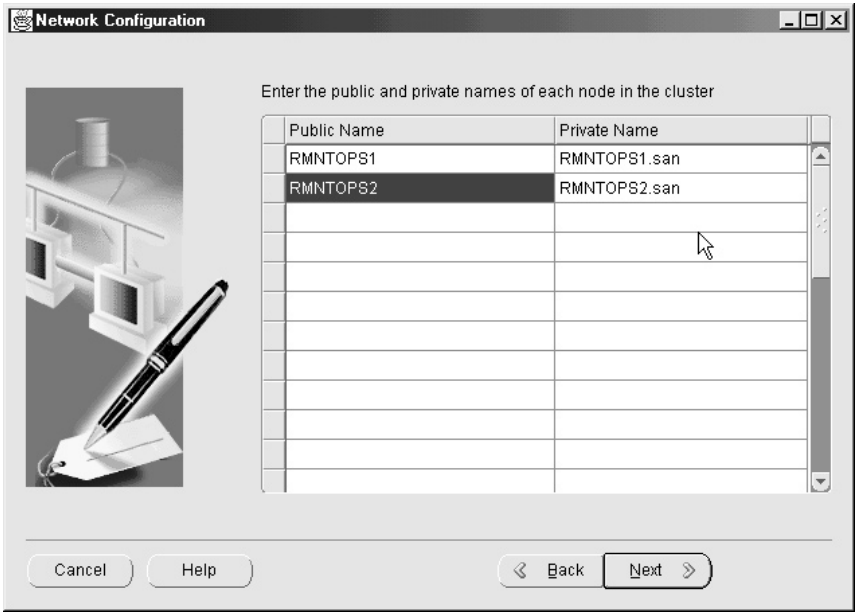
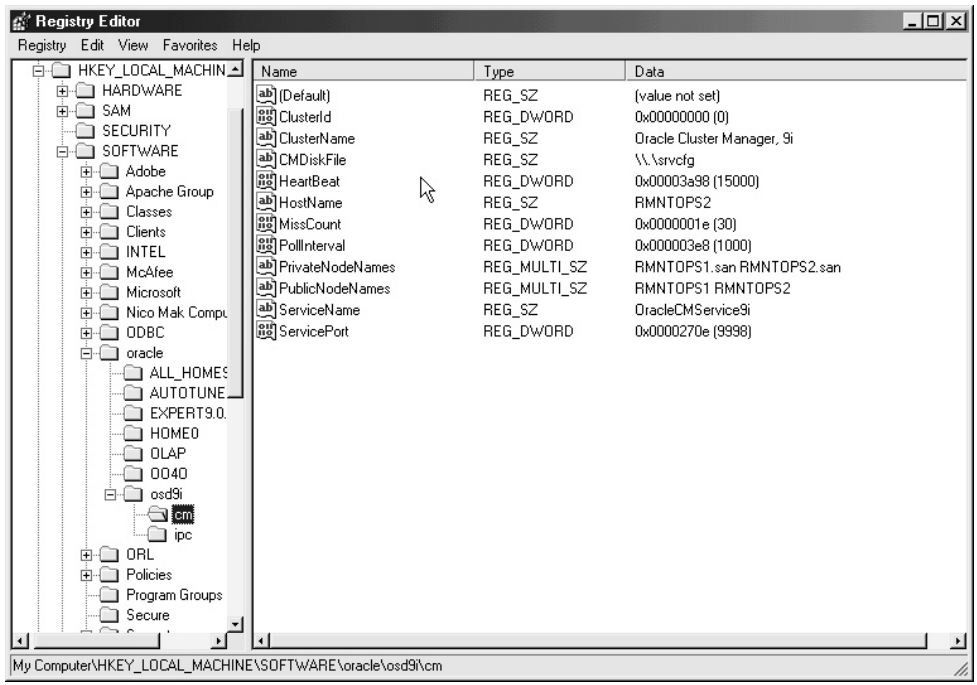


FIGURE 12-9. *Defining the public and private names in cluster setup*

to all nodes defined during the installation. You will see that on each node, an OSD9i Registry key has been created under HKLM\Software\Oracle\osd9i. When highlighted, the subkey called CM will display the current host name, all public node names, all private node names, and the voting disk (as defined by the value for CMDiskFile). In addition, services are created for the Cluster Manager and the Object Link Manager (OracleCMService9i and Oracle Object Service, respectively). Last, observe that the osd9i directory, created in \WINNT\System32 during the installation, is also created on all nodes of the cluster. The log file for the CM server will be located in this directory for troubleshooting purposes. In addition, all of the utilities from the \preinstall_rac\olm directory will be copied to \WINNT\System32\osd9i\olm, as shown next.



Removing or Reinstalling the Cluster Software

If you find that you need to reinstall the cluster software for some reason, you will want to clean up the previous installation first. If instances are running on the machines, you should stop all services for the instances and set them to manual, as you may need to reboot. Next, delete the OSD9i key under HKLM\Software\Oracle, and also delete the services under HKLM\System\CurrentControlSet\Services, OracleCMService9i and OracleGSDService. Last, remove the osd9i directory under \WINNT\System32. After you reinstall the cluster software, you may need to manually re-create the OracleGSDService (which is created initially during the database creation, after the Oracle install completes). Change to the ORACLE_HOME\bin directory and run the following command on each node. If the service does not exist, this command will re-create it:

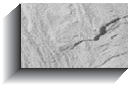
```
D:\oracle\ora90\BIN>gsdservice –start

OracleGSDService
Version 9.0.1

Copyright 1989-2001 Oracle Corporation. All rights reserved.
The service OracleGSDService has been started
```

Installing the Oracle Software

As noted previously, if the cluster software is not installed ahead of time, you will not be given the option to install the option for Real Application Clusters. So, once the cluster software has been successfully installed, you are ready to install Oracle. Be sure that OracleCMService9i is running during the installation, and follow the installation procedures outlined in Chapter 4 when performing the installation from node 1.

Installing from One Location

Choose your ORACLE_HOME with the understanding that the installation is going to copy files to the same location on all nodes. On the Available Products screen, choose the Oracle9*i* Database, and then choose Enterprise Edition for the installation type. Pick the type of database that you want on the Database Configuration screen. Remember that the previously defined links will allow you to create any type of database listed. Even the custom configuration will use the same links unless you specifically modify the link names. The custom configuration has the advantage of allowing you to deselect some of the options, therefore you may not need all of the links and partitions defined earlier. Remember that the global database name that you specify must match the symbolic link names you have defined (in this example, RACDB). You will know that the cluster software is recognized when you go to the next screen after the Database Configuration screen. Here, you should see your public node names listed on the Cluster Node Selection screen, as shown in Figure 12-10. As the wizard states, the current node will always be selected, but you need to ensure that you manually highlight any additional nodes that you want the installation propagated to. From here, go to the Summary screen, and then proceed with the installation.



NOTE

When the installation reaches 100 percent, it will sit there for quite some time, and it will appear to be hung. Do not kill the installer. This is normal behavior, and is due to the fact that the installation is being pushed to the additional nodes in the cluster. How long the installation actually sits at this stage is dependent on the speed of the system. If installing to a cluster with more than two nodes, please obtain the patch for BUG#2031489 from Oracle Support before beginning the install.

Creating the Database During the Installation

At the end of the installation, the Database Configuration Assistant will be started and immediately begin to create the database, using the predefined link names it is programmed to anticipate. If there was a problem creating any of these links, or if

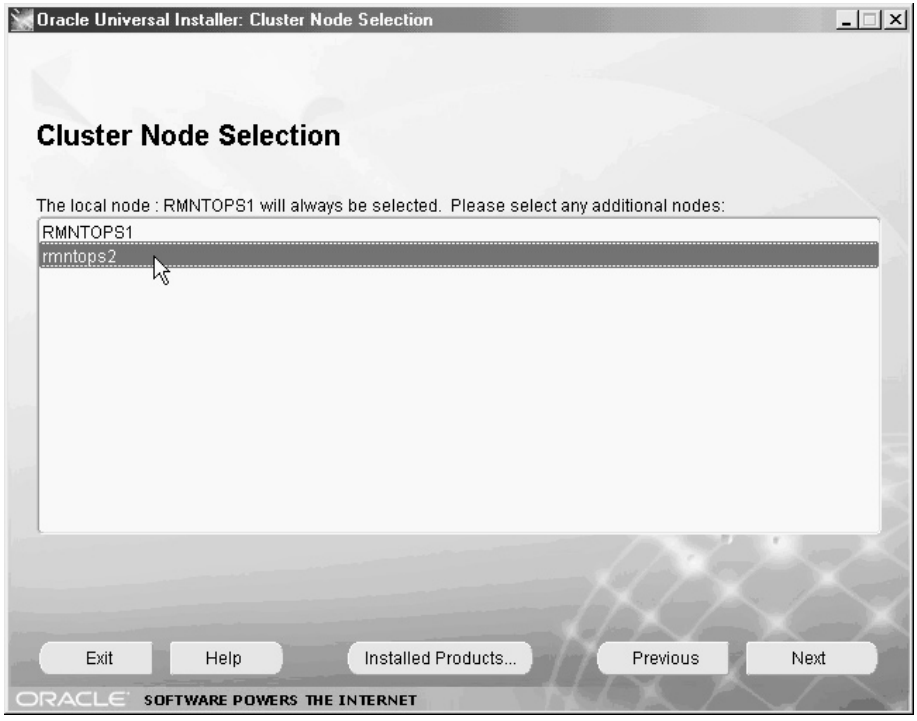


FIGURE 12-10. *Cluster node selection showing all nodes in the cluster*

they were not created, you will see an error during the validation stage, as noted in Figure 12-11.

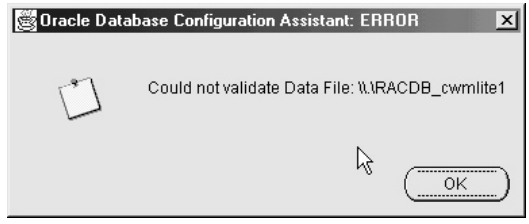
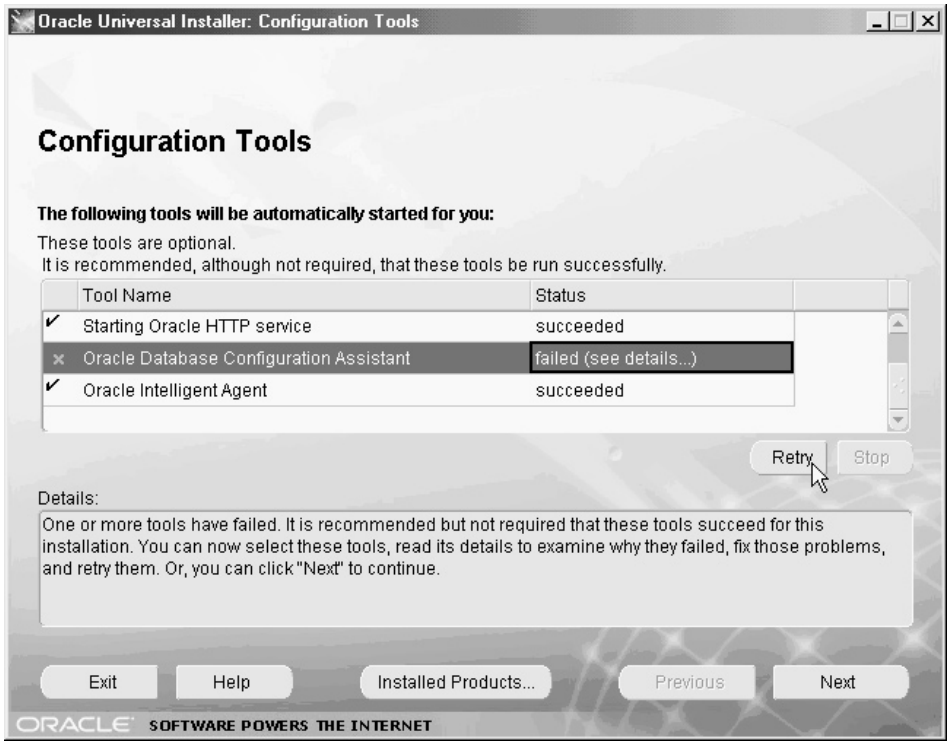


FIGURE 12-11. *Error during file validation phase*

If you receive an error such as this, you can leave the Configuration Tools screen on the installer up, and go back to the Object Link Manager, correcting any mistakes that were made in the link names. Then, go back to the Configuration Tools screen, highlight the Oracle Database Configuration Assistant line, and choose Retry.



Updating the Path on Secondary Nodes

Once the installation is complete, you should be able to observe that all program groups, services, Registry keys, and so on, have been updated on all nodes that were selected for the installation. The path will be updated too; however, we have found that the secondary nodes do not recognize this path change. It is not necessary to reboot the secondary nodes; instead, simply right-click the My Computer icon on the desktop and choose Properties | Advanced. Select Environment Variables, and then you can observe that the path, as defined in System Variables, will have the appropriate Oracle directories already defined. Simply choose OK on this screen, and the path will be updated to reflect the changes previously made by the Oracle Universal Installer.

Creating the Database After the Installation

It is not necessary to create the database during the installation. The Database Assistant can be run at any time. When your cluster software is running, it will be detected and you will be given the option to create either a cluster database or a single instance database. It is possible to have more than one cluster database running, as long as you have properly defined the necessary links using the global database name of the second database. Oracle also supports creation of a nonclustered, or single-instance, database on one or all of the nodes in the cluster.

Additional Considerations in a RAC Environment

Once the database has been created, the view V$ACTIVE_INSTANCES can be queried to determine what instances are currently running. If you want to know which instance you are currently connected to, simply query the V$INSTANCE view. The command-line utility lsnodes can be used at a command prompt to determine which nodes the cluster is active on.

Maintaining a RAC Instance

Just as Oracle Failsafe introduces some vagaries when it comes to maintenance operations, Real Application Clusters provides some twists that one needs to be aware of when operating in a RAC environment. As mentioned previously, the Oracle9*i* documentation set has a total of six separate books on Real Application Clusters, so we cannot be comprehensive here by any stretch. However, we will touch on some of the essentials of which you should be aware.

Services for Multiple Instances

First, be aware that even though the global database name is RACDB (in our example), the instance names on each node will have the node number appended. Thus, node 1 will have an instance name of RACDB1 and a service will be created called OracleServiceRACDB1. Node 2 will have an instance name of RACDB2, with a service name of OracleServiceRACDB2. On node 1, the ORACLE_SID will be set in the Registry to RACDB1, and of course it will be set to RACDB2 on node 2. Likewise, the Listener will be configured differently on each node, with the SID_LIST section containing either RACDB1 or RACDB2, respectively. If you find that you need to manually re-create any of these services or files, be aware of these differences.

Configuration Files

By the same token, the password file in the ORACLE_HOME\Database directory will be named either pwdracdb1.ora or pwdracdb2.ora. This is contrary to a Failsafe setup, where Failsafe maintains an identical password file for each instance. Of course, if you are using OS authentication, this is not an issue.

SPFILE in a RAC Environment The init file is an interesting story as well. If you chose to use the SPFILE during database creation (which is the default), you will find that the initracdb1.ora file, which exists in the ORACLE_HOME\Database directory, will have only one entry, as we recommended in Chapter 5. That entry points to the link name for the SPFILE:

```
SPFILE='\\.\RACDB_spfile1'
```

In prior releases, before the SPFILE came into existence, it was common practice to keep separate parameter files, which included instance-specific configuration information. Then another, common parameter file was maintained that contained parameters that had to be the same across all instances. The SPFILE does away with this, because each instance points to the same logical partition—both instances point to \\.\RACDB_spfile1. There is still the need for instance-specific parameters, but this problem is resolved by prefacing these parameters with instance_name. Here are some example parameters from our SPFILE:

```
cluster_database_instances=2
cluster_database=true
RACDB1.instance_name=RACDB1
RACDB2.instance_name=RACDB2
RACDB1.instance_number=1
RACDB2.instance_number=2
RACDB1.thread=1
RACDB2.thread=2
RACDB1.undo_tablespace=UNDOTBS
RACDB2.undo_tablespace=UNDOTBS2
```

We want to point out a couple of things here. First, notice that the filenames are prefaced with the \\.\<link_name> convention. All files on the raw partitions are referenced in this manner (notice the CONTROL_FILES parameter). Second, observe that the parameter to define a parallel database has changed. In previous releases, this was defined by specifying PARALLEL_SERVER=TRUE. Now, the parameter is CLUSTER_DATABASE=TRUE. In addition, note that the instance name prefaces certain parameters. Last, you should be aware that the initial release of RAC on Oracle9*i* does not support dynamic changes to memory parameters, or the Dynamic SGA, as described in Chapter 5. Therefore, changes to memory parameters must be written to the SPFILE alone, rather than specifying a scope of memory, and the instance must be restarted before changes will take effect.

You can create an init file that can easily be viewed by issuing the **create pfile** command, as discussed in Chapter 5. For example,

```
SQL> create pfile='D:\Oracle\admin\racdb\pfile\initbak.ora' from spfile;
```

Since the SPFILE is on a raw partition, you will need to run this command in order to view its contents.

System-Managed UNDO and/or Rollback in a RAC Environment

As mentioned earlier, each instance must have its own undo tablespace when using system managed undo, so you can see in the previous example that RACDB1 is using UNDOTBS, and RACDB2 is using UNDTOBS2. If you prefer to use the older method of defining standard rollback segments, you can get by with only one tablespace for rollback. However, each instance must define its own rollback segments in the SPFILE or init.ora.

Redo Log Groups

As with system managed undo, each instance must have its own separate set of Redo Log groups, with a minimum of two per instance. With two instances, you will need a total of four groups. When creating groups, thread numbers are assigned to the groups, and those threads are then picked up based on the THREAD=X parameter in the init.ora file. If the need arises to drop these log groups, say, if you were to manually remove a node, you would need to disable the thread:

```
SQL> alter database disable thread 2;
```

Alternatively, a thread must be enabled if adding another node, using the **enable** command.

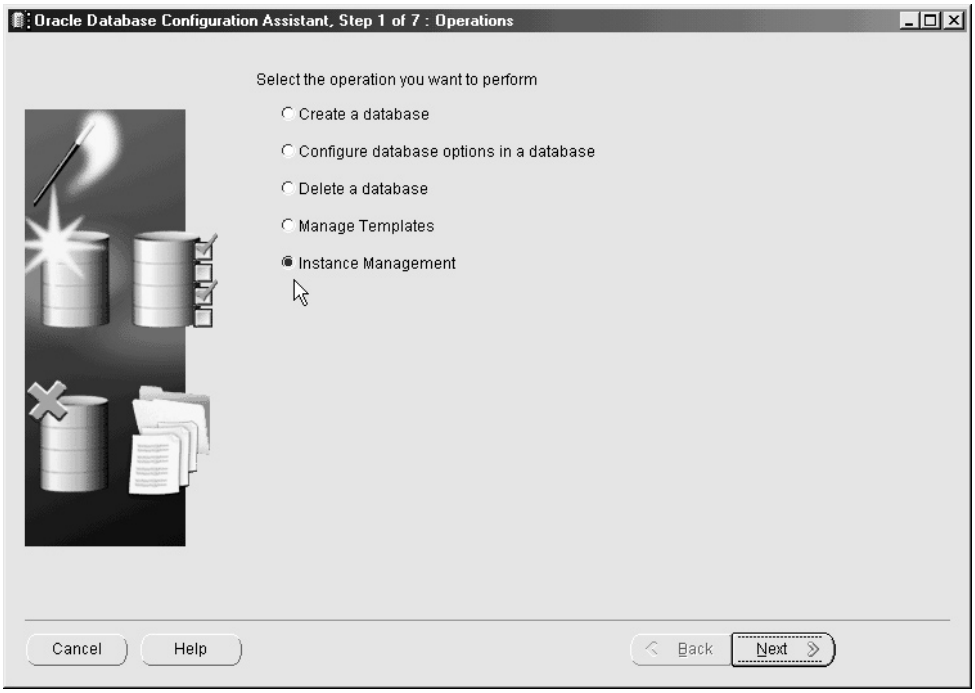
Even though the redo log groups are assigned to a particular instance, they still must be on a raw device, accessible by any instance. In the case of an instance crash, the redo logs for the downed instance will be read by one of the surviving instances, and the automatic instance recovery that would normally occur at startup will instead take place while the surviving instance(s) are running. If this happens, you will notice a pause in the surviving instance(s) during instance recovery.

Archiving with Multiple Threads of Redo

Having multiple redo streams adds some complexity to running in archivelog mode. We discuss this in more detail in Chapter 15; but for our purposes here, you should be aware of a couple of things. First, all archived redo logs must go to a file system; so, generally, archiving is done to the private drives. It is possible to map a network drive to the other node(s) in the cluster, so that a single node is archiving its own private drive, and to the private drive of the other node. Another point of note is that the thread number must always be part of LOG_ARCHIVE_FORMAT, ensuring that each archived redo log will always be created with a unique name. Again, we discuss these issues in more detail in Chapter 15.

Adding Additional Instances

The Database Configuration Assistant simplifies the process of adding additional nodes and instances by providing additional functionality in a RAC environment. Aside from the option to create a clustered database, the DBCA adds an Instance Management feature. You would use this if you have added a node after the installation. Recall that a third or fourth node can be added to an existing cluster by running the Cluster Setup Wizard and choosing the option to add a node. If you add a node to the cluster, you should also run the DBCA, and after choosing the Oracle Cluster Database option, select the option for Instance Management. This will walk you through the process of defining an instance on your new node. If you were adding a third node, it would create an instance named RACDB3, with the associated services and so on, create and enable the thread for the additional instance, and create the undo tablespace for the additional instance. Of course, you must prepare for this by first creating the partitions and assigning the link names using the Oracle Object Link Manager. The Instance Management option also gives you the choice to delete an instance, should the need arise.



Adding Datafiles and Creating Additional Tablespaces

Creating additional tablespaces is a simple enough prospect, now that you are a veteran with the Disk Management Console and the Oracle Object Link Manager. You need to know in advance the size of the partition to create, and carve it out as a logical drive using Disk Management. Remember to delete the drive letter from the additional nodes using the LetterDelete utility. Run the Oracle Object Link Manager (remember that this was installed in the \WINNT\System32\osd9i\olm directory), and define the link name for the partition you have created. Next, simply use the syntax of \\.\<linkname> to add the file to your tablespace (or create the new tablespace):

```
SQL> alter tablespace users add datafile '\\.\RACDB_JUNK' size 100m;
```

Remember that the size you specify must be slightly smaller (by at least 1MB) than the actual size of the partition that you have created.

Using the ocopy Command to Back Up Files on Raw Partitions

Since files cannot be copied from a raw partition using a conventional copy command, and tape devices do not copy directly from a raw device, it is common to use RMAN as the primary backup mechanism of a RAC database. RMAN does not care what type of file system the datafiles are on, since it strictly backs up data blocks. However, if you need to copy files from a raw partition, either for backup purposes (perhaps to create a clone database) or just to be able to view the SPFILE, you can use the **ocopy** command. The syntax for this command is shown here:

```
ocopy from_file [to_file [a | size_1 [size_n]]]
```

In order to copy the SPFILE, for example, the following syntax, using the link name for the SPFILE, will copy it to drive D:

```
D:\oracle\ora90\bin>ocopy \\.\racdb_spfile1 d:\backups\spfile1.ora
```

This will now create a file on drive D: that can be viewed using Wordpad, as a regular SPFILE can. Note that **ocopy** copies the entire raw partition, regardless of how much data is actually used. The rest of the file is just empty filler. Also, be aware that **ocopy** does not copy to tape—if you want to back up files to tape, you must first copy them to a drive on a file system as a staging area, and then back up the files to tape from there.

Summary

This chapter has provided a cookbook approach to implementing Real Application Clusters on a Windows 2000 Cluster. After reading through this chapter, you should have a solid and thorough understanding of how to set up and implement RAC in your environment, from defining your partitions to creating the symbolic links, installing and configuring the cluster software, and installing and creating the database. In Chapter 15, we will discuss further strategies on backing up a RAC database using RMAN, and combining this with a standby environment to provide further levels of availability. In addition, we will discuss combining a RAC setup with the clustering capability of Microsoft Cluster Services, using Real Application Cluster Guard.